

519 8-1(012)

К 03

Комп'ютерне моделювання систем та процесів. Методи обчислень

Частина 1

Вінниця
ВНТУ
2013

Міністерство освіти і науки, молоді та спорту України
Вінницький національний технічний університет

Комп'ютерне моделювання систем та процесів
Методи обчислень
Частина 1

Навчальний посібник

Вінниця
ВНТУ
2013

УДК 519.876.5(075)

ББК 32.97в6я73

К32

Автори:

Р. Н. Кветний, І. В. Богач, О. Р. Бойко, О. Ю. Софіна, О. М. Шушура

Рекомендовано Міністерством освіти і науки, молоді та спорту України як навчальний посібник для студентів вищих навчальних закладів, які навчаються за напрямом підготовки «Системна інженерія». Лист № 1/11-1254 від 01.02.2012 р.

Рецензенти:

Б. П. Русин, доктор технічних наук, професор

Г. С. Фінін, доктор фізико-математичних наук, ст.н.сп.

А. М. Петух, доктор технічних наук, професор

К32

Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 1 : навчальний посібник / [Р. Н. Кветний, І. В. Богач, О. Р. Бойко та інші]; за заг. ред. Р. Н. Кветного. – Вінниця : ВНТУ, 2013. – 191 с.

ISBN 978-966-641-520-5 (частина 1)

Перша частина навчального посібника, в якому розглянуто найпоширеніші чисельні методи, що зустрічаються в типових інженерних та наукових задачах, методи оптимізації та основи математичного моделювання, а також методи цифрової обробки сигналів та зображень, фрактальний та інтервальний аналіз. Призначено для студентів напряму підготовки «Системна інженерія» при вивченні дисципліни «Комп'ютерне моделювання систем та процесів», але може бути використано при вивченні широкого спектру дисциплін цього та інших напрямів, які пов'язані з комп'ютерними обчисленнями та обробкою даних, сигналів, зображень, а також для наукової роботи студентів, аспірантів, інженерів та вчених. Наведено широкий спектр прикладів та задач.

УДК 519.876.5(075)

ББК 32.97в6я73

ISBN 978-966-641-519-9 (загальний)

ISBN 978-966-641-520-5 (частина 1)

© Р. Кветний, І. Богач, О. Бойко, О. Софіна, О. Шушура, 2013

ЗМІСТ

| | |
|--|----|
| Вступ | 6 |
| РОЗДІЛ 1 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ | 8 |
| 1.1 Поняття моделі та моделювання. Властивості та класифікація моделей | 8 |
| 1.2 Узагальнена методика математичного моделювання | 12 |
| 1.3 Аналітичне моделювання..... | 16 |
| 1.3.1 Особливості аналітичного динамічного моделювання.... | 16 |
| 1.3.2 Особливості аналітичного статичного моделювання..... | 19 |
| 1.4 Комп'ютерне та імітаційне моделювання | 22 |
| 1.4.1 Алгоритми і програмування | 22 |
| 1.4.2 Особливості комп'ютерного моделювання | 27 |
| 1.4.3 Особливості імітаційного статистичного моделювання... | 28 |
| 1.5 Похибки та властивості обчислювальних алгоритмів | 40 |
| 1.6 Математичне моделювання і обчислювальні методи в задачах проектуювання комп'ютерних систем автоматики та управління | 43 |
| 1.7 Інструментальні засоби моделювання..... | 45 |
| 1.7.1 Моделювання з використанням математичних пакетів... | 45 |
| 1.7.2 Програмні пакети для імітаційного моделювання..... | 48 |
| 1.7.3 Системи автоматизованого проектування..... | 56 |
| Контрольні запитання та завдання | 62 |
| РОЗДІЛ 2 ЗАДАЧІ ЛІНІЙНОЇ АЛГЕБРИ | 64 |
| 2.1 Розв'язання систем лінійних рівнянь | 64 |
| 2.1.1 Прямі методи | 65 |
| 2.1.1.1 Метод Гаусса | 65 |
| 2.1.1.2 Метод виключення Гаусса-Жордана | 69 |
| 2.1.1.3 Модифікований метод Гаусса | 69 |
| 2.1.1.4 Застосування прямого ходу метода Гаусса для пошуку визначників | 71 |
| 2.1.1.5 Метод Крамера | 72 |
| 2.1.1.6 Метод оберненої матриці | 73 |
| 2.1.1.7 Метод прогонки | 73 |
| 2.1.2 Ітераційні методи | 74 |
| 2.2 Визначення власних значень матриць | 77 |
| 2.2.1 Постановка проблеми | 77 |
| 2.2.2 Методи обчислення власних значень | 78 |
| 2.2.2.1 Прямі методи | 79 |

| | |
|---|------------|
| 2.2.2.2 Ітераційні методи | 80 |
| 2.2.2.3 Методи перетворень подібності | 81 |
| 2.2.3 Порівняння методів визначення власних значень | 83 |
| Контрольні запитання та завдання | 84 |
| РОЗДІЛ 3 НЕЛІНІЙНІ ЗАДАЧІ | 87 |
| 3.1 Розв'язання нелінійних рівнянь | 87 |
| 3.1.1 Метод половинного ділення | 91 |
| 3.1.2 Метод хибного положення (хорд) | 92 |
| 3.1.3 Метод Ньютона (дотичних) | 95 |
| 3.1.4 Метод січних | 97 |
| 3.1.5 Метод простої ітерації | 98 |
| 3.1.6 Визначення комплексних коренів | 99 |
| 3.2 Розв'язання систем нелінійних рівнянь | 101 |
| Контрольні запитання та завдання | 106 |
| РОЗДІЛ 4 МЕТОДИ РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ | 108 |
| 4.1 Методи розв'язання задачі Коші | 110 |
| 4.1.1 Однокрокові методи | 111 |
| 4.1.2 Багатокрокові методи | 118 |
| 4.1.3 «Жорсткі» задачі | 121 |
| 4.1.4 Вибір методу розв'язання задачі Коші | 121 |
| 4.2 Методи розв'язання крайових задач | 122 |
| 4.2.1 Метод «стрілянини» | 123 |
| 4.2.2 Різницеві методи | 123 |
| Контрольні запитання та завдання | 126 |
| РОЗДІЛ 5 МЕТОДИ РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ В ЧАСТИННИХ ПОХІДНИХ | 127 |
| 5.1 Різницевий метод | 129 |
| 5.2 Розв'язання різних типів диференціальних рівнянь в частинних похідних | 133 |
| 5.2.1 Еліптичні рівняння | 133 |
| 5.2.2 Гіперболічні рівняння | 135 |
| 5.2.3 Параболічні рівняння | 136 |
| 5.3 Загальні рекомендації до розв'язання диференціальних рівнянь в частинних похідних | 137 |
| Контрольні запитання та завдання | 138 |

| | |
|---|-----|
| РОЗДІЛ 6 МЕТОДИ ОБРОБКИ ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ | 139 |
| 6.1 Інтерполяція | 139 |
| 6.1.1 Різницеві методи | 140 |
| 6.1.2 Інтерполяція за Лагранжем | 147 |
| 6.1.3 Сплайн-інтерполяція | 149 |
| 6.1.3.1 Класичний кубічний сплайн | 152 |
| 6.1.3.2 Створення сплайнових параметричних кривих | 155 |
| 6.2 Апроксимація даних | 161 |
| 6.3 Статистична обробка даних | 164 |
| 6.4 Чисельне інтегрування | 168 |
| 6.4.1 Формули прямокутників | 169 |
| 6.4.2 Формули Ньютона -Котеса | 171 |
| 6.4.3 Формула Чебишева | 173 |
| 6.4.4 Формула Гаусса | 175 |
| 6.4.5 Оцінка похибки при чисельному інтегруванні | 178 |
| 6.4.6 Алгоритми застосування чисельних методів | 178 |
| 6.4.7 Метод Монте-Карло..... | 180 |
| 6.5 Чисельне диференціювання | 183 |
| 6.5.1 Чисельне диференціювання аналітично заданих функцій | 183 |
| 6.5.2 Чисельне диференціювання таблично заданих функцій.. | 185 |
| Контрольні запитання та завдання | 186 |
| Література | 188 |

ВСТУП

Одним з головних напрямків науково-технічного прогресу протягом вже кількох десятиріч є розвиток методів і засобів інформатики та обчислювальної техніки. Використання методів математичного моделювання та комп'ютерного розв'язання інженерних і наукових задач дозволяє значно підвищити ефективність процесів проектування та управління. Впровадження персональних комп'ютерів, комп'ютерних інформаційних мереж, побудова та розвиток INTERNET, широке та різноманітне використання методів математичного моделювання привели до розширення як практичної, так і теоретичної баз комп'ютерної математики. Математичне комп'ютерне моделювання стало головним засобом дослідження складних процесів і систем, на якому базуються сучасні підходи до проектування, оптимізації та управління в різних галузях науки і техніки. Обчислювальна математика стала основою для реалізації та комп'ютерного розрахунку методів математичного моделювання.

Метою цієї книги, що видана в двох частинах, є ознайомлення широкого кола студентів, науковців, інженерно-технічних працівників з основними поняттями комп'ютерного моделювання систем і процесів та методами розв'язання на комп'ютерах сучасних задач обчислювальної математики, що виникають в процесі дослідження й проектування систем управління та автоматики.

Посібник призначений для студентів напрямів підготовки «Системна інженерія» та «Комп'ютерна інженерія», але також може бути використаний для інших спеціальностей при вивченні дисциплін, пов'язаних з чисельними методами, методами обробки даних, оптимізацією та плануванням експерименту.

Теоретичною основою посібника стали відомі роботи в області моделювання та обчислювальної математики Л. Коллатця, Дж. Форсайта, Р. Мура, Д. Кнута, А. Самарського, Б. Демідовича, І. Марона, В. Скурихіна, М. Бусленка, А. Крилова, А. Верлани, Г. Марчука та інших вітчизняних і закордонних вчених. В посібнику узагальнено досвід багаторічного викладання Р. Н. Кветним курсів з обчислювальної математики та моделювання, використано результати наукових і навчально-методичних розробок співавторів цієї книги.

Перший розділ першої частини книги присвячений розгляду основних понять і підходів до побудови математичних комп'ютерних моделей. В наступних розділах розглянуто найбільш поширені задачі обчислювальної математики, методи та алгоритми їх розв'язання. Ці задачі розглянуто в першій частині навчального посібника. Друга частина присвячена задачам обробки сигналів та зображень, що дуже поширені в практиці комп'ютерного моделювання, інтервальним методам та ще деяким

проблемам, які пов'язані з розрахунками в процесі математичного моделювання й можуть бути корисними для сучасних науковців та інженерів.

Використання посібника передбачає знання основ програмування, а алгоритмізація та реалізація комп'ютерних програм може здійснюватись з застосуванням будь-якої зручної для користувача мови програмування. В книзі наведено приклади розв'язання деяких задач з використанням процедур програмних систем MathCad, MATLAB та інших, але не дається загальних основ використання цих систем, що не входить до задач цього посібника. Автори намагалися зробити так, щоб викладений матеріал не дуже підпадав під вплив часу, тому робили наголос на висвітленні базових понять. Особливу увагу приділено алгоритмізації обчислювальних процедур як основ застосування комп'ютерної математики.

Розділ 1 першої частини підготовлено Р. Н. Кветним, О. Ю. Софиною, О. М. Шушурою; розділи 2, 3 – Р. Н. Кветним, І. В. Богач, О. М. Шушурою; розділи 4, 5 – Р. Н. Кветним; розділ 6 – Р. Н. Кветним та О. М. Шушурою. В другій частині розділи 1, 2 – О. Ю. Софиною (2.2 – разом з І. В. Богач); розділ 3 – О. Р. Бойко; розділ 4 – Р. Н. Кветним. Загальну редакцію книги здійснено Р. Н. Кветним.

РОЗДІЛ 1 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

1.1 Поняття моделі та моделювання. Властивості та класифікація моделей

Натурний експеримент (Natural experiment), тобто дослідження властивостей та поведінки об'єкта управління в певних умовах з використанням самого об'єкта, є важливою складовою у сферах проектування та управління. Однак у багатьох випадках натурне моделювання є неможливим або недоцільним. Наприклад, експерименти на об'єкті керування при управлінні технологічними процесами у режимі реального часу, проектуванні складних систем та пристроїв можуть бути економічно недоцільні або неможливі через неготовність самого об'єкта.

Модель (від лат. *modulus* – міра, зразок, норма) – це об'єкт-замінник, створений з метою відтворення за певних умов суттєвих властивостей об'єкта-оригіналу. Модель може бути подана фізичним об'єктом, подібним до оригіналу, або описом об'єкта у вигляді математичних формул, тексту, комп'ютерної програми.

Основним призначенням моделі в задачах управління є прогноз реакції об'єкта на керувальні впливи. Крім того, моделі використовуються для дослідження об'єкта, аналізу його чутливості.

Основні властивості моделей:

- цілеспрямованість;
- скінченність;
- спрощеність;
- повнота;
- адекватність.

Цілеспрямованість моделі полягає в тому, що вона завжди будується з певною метою. Ця мета впливає на те, які властивості об'єктивного явища вважаються істотними, а які – ні. Модель є, так би мовити, проекцією об'єктивної реальності під певним кутом зору. Наприклад, моделі вищого навчального закладу як інформаційної, фінансової, енергетичної та соціальної систем будуть зовсім різними. Інколи, залежно від мети, можна отримати ряд проєкцій об'єктивної реальності, що вступають у протиріччя. Це характерно, як правило, для складних систем, в яких кожна проєкція виділяє суттєве для певної мети з безлічі несуттєвого. *Задача моделювання* полягає в тому, що для заданого об'єкта потрібно підібрати такий опис, який у повній мірі відобразив би оригінал з точки зору заданої мети моделювання.

Скінченність моделі визначає те, що модель відтворює лише скінченну кількість властивостей та відношень, і через це модель завжди є більш простою, ніж оригінал.

Повнота моделі полягає в тому, що вона має відображати всі істотні, з точки зору мети моделювання, властивості оригіналу.

Необхідною умовою для переходу від дослідження об'єкта до дослідження моделі і подальшого перенесення результатів на об'єкт дослідження є вимога адекватності моделі і об'єкта. *Адекватність* – це відтворення моделлю з необхідною повнотою всіх властивостей об'єкта, важливих для цілей даного дослідження. Це, мабуть, найголовніша властивість моделі, яка визначає можливість її використання. Оскільки будь-яка модель простіша за оригінал, ніколи не можна говорити про абсолютну адекватність, при якій модель за всіма характеристиками відповідає оригіналу. Модель називається *ізоморфною* (однаковою за формою), якщо між нею і реальною системою існує повна поелементна відповідність, і *гомеоморфною*, якщо існує відповідність лише між найбільш значними складовими частинами об'єкта і моделі.

Моделювання (Modeling) охоплює створення, дослідження та використання моделей об'єктів. Методи моделювання широко використовуються в різних сферах людської діяльності, особливо в сферах проектування та управління, де основними є процеси ухвалення ефективних рішень на основі отримуваної інформації. *Метою моделювання* є здобуття, обробка, подання і використання інформації про об'єкти, які взаємодіють між собою і зовнішнім середовищем; а модель тут виступає як засіб пізнання властивостей і закономірностей поведінки об'єкта.

Теорія моделювання є розділом науки, що вивчає способи дослідження властивостей об'єктів (оригіналів) на основі заміщення їх іншими об'єктами (моделями). Вирізняють натурні, фізичні, мовні та математичні моделі.

Зупинимось на одному з найбільш універсальних видів моделювання – математичному, що ставить у відповідність модельованому фізичному процесу систему математичних співвідношень, розв'язання якої дозволяє отримати відповідь на питання про поведінку об'єкта без створення фізичної моделі, яка часто є дорогою і малоефективною. Отже, *математичною моделлю* називається сукупність математичних співвідношень, рівнянь, нерівностей, що описують основні закономірності, властиві досліджуваному процесу, об'єкту або системі.

На підставі різних критеріїв класифікації виділяють такі види моделей:

- динамічні або статичні;
- детерміновані або стохастичні;
- неперервні, дискретні або дискретно-неперервні;
- лінійні чи нелінійні;
- з розподіленими або зосередженими параметрами;
- аналітичні, імітаційні чи комп'ютерні.

Динамічні моделі (dynamic models) відтворюють поведінку нестационарних об'єктів, що змінюються у часі. Статичні моделі описують стан об'єкта у деякий момент часу. Такі моделі розробляються для стаціонарних об'єктів, зміни яких у часі не є істотними стосовно періоду розробки та використання моделі.

Детерміновані моделі (deterministic models) використовують для опису процесів, що не містять істотної випадковості. Наприклад, поведінку більшості технічних систем можна охарактеризувати за допомогою так званих фазових змінних – фізичних величин типу потоку і потенціалу. При цьому доцільно виділити в об'єктах моделювання досить великі елементи, що розглядаються як невідимі одиниці. Закони функціонування елементів системи задаються компонентними рівняннями, що зв'язують різномірні фазові змінні. Загальність опису процесів, що відбуваються в різних технічних системах, дозволяє виділити декілька типів елементів: R – елемент розсіювання енергії; C і L – елементи накопичення енергії. Поєднанням цих простих елементів і джерел фазових змінних отримують еквівалентну схему технічної системи будь-якої складності та її математичну модель. Конкретний зміст фазових змінних і простих елементів фізичних систем наведений у таблиці 1.1.

Таблиця 1.1 – Зміст фазових змінних і простих елементів фізичних систем

| Підсистема | Фазові змінні | | Елементи | | |
|---------------------------|----------------|------------------|-----------|---------------------|---------------------------|
| | типу течії | типу потенціалу | типу R | типу C | типу L |
| Електрична | Струм | Напруга | Опір | Ємність | Індуктивність |
| Механічна поступальна | Сила | Швидкість | Тертя | Маса | Пружність |
| Механічна обертальна | Момент | Кутова швидкість | Тертя | Момент інерції | Обертальна гнучкість |
| Гідравлічна (пневматична) | Витрата | Тиск | Тертя | Гідравлічна ємність | Гідравлічна індуктивність |
| Теплова | Тепловий струм | Температура | Теплоопір | Теплоємність | - |

Для моделювання нестационарних імовірнісних процесів використовують стохастичні моделі (stochastic models). Якщо об'єкт моделювання стаціонарний і піддається випадковим впливам, то модель називають статистичною. Наприклад, для моделювання функцій перетворення вимірювальних пристроїв досить скористатися детермінованим способом опису, тоді як для аналізу похибок, оцінювання інформаційних характеристик необхідно застосувати ймовірнісно-статистичні методи.

Неперервні моделі (continuous model) описують системи з неперервними процесами, а дискретні моделі відображають поведінку

систем з дискретними станами. Дискретно-неперервні моделі використовуються, коли на об'єкті виділяються обидва типи процесів.

Якщо при описі моделі використовуються лише лінійні математичні конструкції (наприклад, лінійні алгебраїчні рівняння), то модель називають лінійною, інакше – нелінійною.

Моделі з розподіленими параметрами (models with distributed parameters) описують просторове поширення явищ, а моделі з зосередженими параметрами нехтують просторовою складовою. Динамічні неперервні детерміновані моделі з розподіленими параметрами використовують апарат диференціальних рівнянь у частинних похідних, а з зосередженими параметрами – звичайних диференціальних рівнянь.

Для аналітичних моделей (analytical models) властиво те, що процеси функціонування об'єкта подаються у вигляді аналітичних математичних залежностей: алгебраїчних, диференціальних, інтегральних рівнянь або їх систем, логічних умов. Наприклад, закон Ома чи рівняння Максвелла. Дослідження аналітичних моделей можливе за допомогою методів:

- аналітичних;
- чисельних;
- якісних.

Аналітичні методи полягають у пошуку явних залежностей між характеристиками. Однак такі залежності можна отримати лише для невеликої кількості простих моделей, як правило, лінійних. Інколи виконують спрощення моделей для отримання можливості вивчити хоча б загальні властивості об'єкта.

Чисельні методи (numerical methods) дозволяють отримати розв'язок аналітичних моделей, для котрих застосування аналітичних методів неможливо або недоцільно. Розв'язання чисельними методами здійснюється для конкретних вихідних даних і має додаткову похибку. Детальніше питання похибок чисельних методів розглядається у підрозділі 1.5.

Якісні методи дозволяють зробити певні висновки стосовно моделі, не маючи розв'язку у явному вигляді. Наприклад, такі методи використовуються у теорії автоматичного управління для оцінювання ефективності різних варіантів систем управління.

Імітаційне моделювання (simulation) передбачає подання моделі у вигляді алгоритму та комп'ютерної програми, яка дозволяє відтворити поведінку об'єкта. *Імітаційні моделі* розглядаються як експерименти, що проводяться на комп'ютерах з математичними моделями, які імітують поведінку реальних об'єктів. При цьому імітуються елементарні явища, що складають процес, зі збереженням їх логічної структури та послідовності у часі, що дозволяє отримати відомості про стан системи у певний момент часу та оцінити характеристики системи. Імітаційні моделі дозволяють

вирішувати більш складні задачі, ніж аналітичні. Наприклад, вони дозволяють досить легко врахувати вплив випадкових факторів.

Традиційно під моделюванням на ЕОМ розумілося лише імітаційне моделювання. Але завдяки розвитку графічного інтерфейсу та графічних пакетів значного поширення набуло комп'ютерне структурно-функціональне моделювання, а також розпочалося використання комп'ютера з метою концептуального моделювання, наприклад для побудови систем штучного інтелекту.

Під комп'ютерною моделлю (computer model) найчастіше розуміють:

– умовний образ об'єкта чи деякої системи об'єктів (або процесів), описаних за допомогою взаємозалежних комп'ютерних таблиць, схем, діаграм, графіків, рисунків, анімаційних фрагментів, гіпертекстів і т. ін., що відбивають структуру та взаємозв'язки між елементами об'єкта чи системи. Комп'ютерні моделі такого типу називають структурно-функціональними;

– окрему програму, сукупність програм чи програмний комплекс, що дає змогу виконанням послідовності обчислень з подальшим графічним відображенням їх результатів відтворювати (імітувати) процеси функціонування об'єкта (системи об'єктів), що функціонує під впливом різних, як правило, випадкових, факторів.

Інколи застосовується комбіноване (аналітико-імітаційне) моделювання, яке полягає в тому, що об'єкт декомпозується на окремі підсистеми. Для тих підсистем, для яких це можливо, використовуються аналітичні моделі, а для інших розробляються імітаційні моделі.

Розробка моделей поєднує в собі науку і мистецтво. На жаль, немає чіткого формального алгоритму, який би дозволив побудувати модель для будь-якого об'єкта. Тому далі розглядаються лише певні методичні рекомендації щодо розробки моделей.

1.2 Узагальнена методика математичного моделювання

Математичне моделювання можна розглядати як засіб вивчення реальної системи шляхом її заміни зручнішою для експериментального дослідження системою (моделлю), що зберігає істотні риси оригіналу. При моделюванні здійснюється апроксимація функції опису більш простою і зручною для практичного аналізу функцією – моделлю.

Математичні моделі, особливо ті, що використовують чисельні методи, потребують для свого створення значних інтелектуальних, фінансових і часових затрат. Тому рішення про створення нової моделі приймається лише в разі відсутності більш простих шляхів вирішення поставленої проблеми (наприклад, модифікації однієї з існуючих моделей).

Дослідження об'єкта моделювання та складання його математичного опису полягає у встановленні зв'язків між характеристиками процесу, виявленні його граничних і початкових умов та формалізації процесу у вигляді системи математичних співвідношень.

Процес побудови будь-якої математичної моделі можна подати послідовністю етапів, зображених на рис. 1.1.

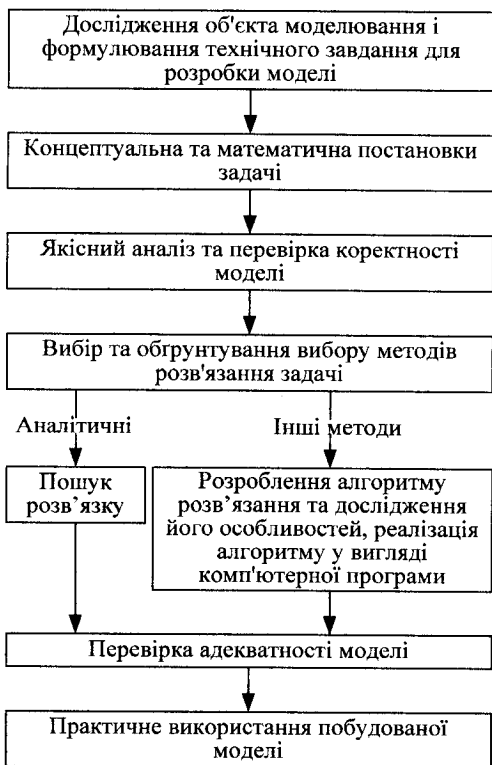


Рисунок 1.1 – Послідовність етапів побудови математичної моделі

На етапі *дослідження об'єкта моделювання* потрібно виконати такі дії:

- аналіз взаємодії об'єкта з зовнішнім середовищем, виділення характеристик вхідних впливів та реакції об'єкта, класифікація їх на вимірні та невимірні, керувані та перешкоди;
- проведення декомпозиції та дослідження внутрішньої структури об'єкта;
- дослідження порядку функціонування об'єкта, виявлення зв'язку між входом та виходом, формування множини станів об'єкта;

- збір та перевірка існуючих експериментальних даних про об'єкти-аналоги, проведення, за необхідності, додаткових експериментів;
- класифікація об'єкта моделювання на стаціонарний чи нестаціонарний, визначення міри впливу випадкових факторів на об'єкт та порядку нелінійності зв'язків між характеристиками об'єкта;
- аналітичний огляд літературних джерел, аналіз та порівняння побудованих раніше моделей подібних об'єктів;
- аналіз та узагальнення всього накопиченого матеріалу, розроблення загального плану створення математичної моделі.

В деяких випадках дослідження внутрішньої будови та порядку функціонування об'єкта моделювання неможливе або економічно недоцільне. Тому можна розглядати об'єкт як «чорний ящик», стосовно якого нам відомі лише його входи та виходи.

На підставі аналізу об'єкта моделювання формується змістовна постановка моделювання, в якій мають бути зазначені:

- мета моделювання;
- тип моделі;
- вимоги до адекватності моделі та якості розв'язку.

Тип моделі має відповідати результатам класифікації об'єкта моделювання, інакше модель навряд чи буде адекватною.

Весь накопичений в результаті дослідження матеріал, змістовна постановка задачі моделювання, додаткові вимоги до реалізації моделі, оформляються у вигляді технічного завдання на проектування та розробку моделі.

Концептуальна постановка задачі моделювання – це сформульований в термінах конкретних дисциплін (фізики, хімії, біології тощо) список основних питань, а також сукупність гіпотез відносно особливостей та поведінки об'єкта моделювання. Розробник моделі на підставі результатів аналізу об'єкта моделювання формує своє бачення стосовно процесів на об'єкті і формулює його природною мовою в термінах предметної області. При цьому з метою спрощення моделі він приймає низку припущень та обмежень. Припущення можуть містити нехтування певними процесами або зміну характеру їх протікання. Концептуальна модель має пройти погодження з експертами у даній предметній галузі з метою перевірки на адекватність. Адекватність концептуальної моделі визначає адекватність математичної моделі, яка формується на її основі.

Математична постановка задачі моделювання – це сукупність математичних співвідношень, які описують поведінку та характеристики об'єкта моделювання. Необхідно формалізувати змінні моделі та зв'язки між ними. Математичний опис моделі складається на основі законів фізики, хімії тощо, які характеризують динаміку і статику процесів в досліджуваному об'єкті, та виражається мовою будь-яких розділів математики. Найбільше поширення при побудові детермінованих моделей

набули алгебраїчні рівняння та системи, звичайні диференціальні рівняння і диференціальні рівняння в частинних похідних, матрична алгебра, а при стохастичному моделюванні – і методи теорії імовірності, математичної статистики та теорії випадкових процесів. Якщо апріорні відомості про об'єкт недостатні, вигляд математичних моделей уточнюється за допомогою методів багатовимірної статистики: регресійного, кореляційного, багатофакторного та інших аналізів, а також проведення пасивного або планування активного експериментів. Для більшості складних об'єктів структура моделі містить параметри, які відображають особливості конкретних об'єктів. Пошук значень цих параметрів відбувається за допомогою методів параметричної ідентифікації на основі проведення пасивного або активного експериментів.

Поняття коректності задачі має важливе значення в процесі моделювання. Адже, наприклад, чисельні методи розв'язання задач доцільно застосовувати лише до коректно поставлених задач. При цьому, не всі практичні задачі можна вважати коректними. Математична модель є *коректною*, якщо для неї отримано позитивний результат у всіх контрольних перевірках: розмірності, порядку, характеру залежностей, граничних умовах, фізичному сенсі тощо.

Для математичної моделі обирається один з методів розв'язання, який дозволяє при заданих значеннях вхідних змінних отримати значення вихідних змінних. Вибір методу обґрунтовується на підставі властивостей моделі, даних про точність вимірювання значень змінних, вимог до точності та швидкості отримання розв'язку.

Необхідною умовою для переходу від дослідження об'єкта до дослідження моделі і подальшого перенесення результатів на об'єкт моделювання є вимога адекватності моделі об'єкта. *Адекватність* – це відтворення моделлю з необхідною повнотою всіх властивостей об'єкта, важливих для цілей даного дослідження. Як правило, адекватність моделі визначається на підставі статистичних оцінок розбіжностей значень вихідних змінних моделі та об'єкта при однакових значеннях вхідних змінних, розрахованих за результатами серії експериментів на об'єкті моделювання. Для перевірки адекватності моделі використовуються дані іншої серії експериментів, ніж для параметричної ідентифікації. Відмінність значень виходу моделі та об'єкта може бути зумовлена такими причинами:

- спрощеність моделі;
- похибка чисельних методів;
- похибка вимірювальних пристроїв;
- обчислювальна похибка, пов'язана з переходом від десяткової до двійкової системи числення та особливостями комп'ютерних обчислень.

Якщо модель не задовольняє критерії адекватності, необхідно крок за кроком перевірити коректність розробки на всіх етапах:

- умови проведення експерименту та правильність вимірювання і фіксування його результатів;
- правильність програмної реалізації алгоритмів;
- адекватність результатів параметричної ідентифікації;
- обґрунтованість вибору методу розв'язання моделі;
- коректність математичного опису явищ та характеристик об'єкта;
- адекватність концептуальної моделі.

Після успішної перевірки адекватності модель може бути застосована в задачах прогнозування та дослідження об'єкта.

Метод математичного моделювання дозволяє уникати необхідності виготовлення громіздких фізичних моделей, пов'язаної з матеріальними витратами; скоротити час визначення характеристик (особливо при розрахунку математичних моделей на комп'ютері і застосуванні ефективних обчислювальних методів і алгоритмів); вивчати поведінку об'єкта моделювання при різних значеннях параметрів; аналізувати можливість застосування різних елементів; отримувати характеристики і показники, які складно отримувати експериментально (кореляційні, частотні, параметричної чутливості).

Розглянемо особливості аналітичного та імітаційного моделювання.

1.3 Аналітичне моделювання

1.3.1 Особливості аналітичного динамічного моделювання

Дана методика застосовується для створення моделей об'єктів, що змінюються в часі (нестационарних об'єктів). Специфіка динамічного моделювання полягає в особливій побудові концептуальної моделі (гіпотези про механізм процесів) і формалізації залежностей математичної постановки.

Концептуальна модель ґрунтується на розгляді об'єкта на кількох рівнях декомпозиції. Як правило, виділяють явища на макро- і мікрорівні об'єкта. На макрорівні процеси розглядаються узагальнено, «великими мазками». Наприклад, рух потоків маси і енергії, надходження грошових коштів і т. п. На мікрорівні ці процеси розглядаються більш детально.

На кожному рівні декомпозиції виділяють явища і зв'язки між ними. При динамічному моделюванні під явищем будемо розуміти зміну деякої величини. У концептуальній моделі описується перебіг кожного явища. Зв'язки між явищами поділяються на взаємозв'язки явищ одного рівня декомпозиції і зв'язки між явищами різних рівнів декомпозиції. Наприклад, для гіпотези з двома рівнями декомпозиції можливі зв'язки

моменти часу, точність), вимоги до швидкості процедури (якщо ідентифікація відбувається в контурі управління).

Формальна постановка задачі ідентифікації полягає в записі фізичної постановки у вигляді задачі оптимізації, головною складовою якої є критерій (функціонал) похибки $E(\bar{\beta})$. Параметри можуть розглядатись і як константи, і як функції. Вигляд критерію похибки залежить від специфіки конкретної задачі. Вирізняють парні та непарні критерії похибки. Непарні критерії залежать від знака відхилення вихідних змінних моделі від змінних об'єкта, а парні – ні. Як приклад розглянемо парний критерій, який призначений для контролю похибки на всьому інтервалі прогнозу $[t_0, T]$:

$$E(\bar{\beta}) = \sum_{j=1}^m \sum_{i=1}^n \int_{t_0}^T (y_i(t, \bar{x}_j, \bar{\beta}) - y_{ij}^*(t, \bar{x}_j))^2 dt \rightarrow \min_{\bar{\beta}},$$

де m – кількість експериментів; n – кількість вихідних змінних, $y_{ij}^*(t, \bar{x}_j)$ – значення змінної об'єкта $y_i^*(t, \bar{x})$ за результатами j -го експерименту.

Звичайно вимірювання проводяться в дискретні моменти часу, тому інтеграли подаються у вигляді дискретних конструкцій за допомогою формул чисельного інтегрування.

В залежності від вигляду критерію $E(\bar{\beta})$ та вимог до швидкості процедури обирається аналітичний чи чисельний методи параметричної ідентифікації. Ці методи спираються на методи оптимізації і, як правило, пов'язані з пошуком частинних похідних $\frac{\partial E}{\partial \beta_i}$. Коли аналітичний вигляд

залежності $y_i(t, \bar{x}, \bar{\beta})$ невідомий, для розрахунку частинних похідних $\frac{\partial E}{\partial \beta_i}$ застосовуються формули чисельного диференціювання. За необхідності, на основі обраного методу параметричної ідентифікації розробляється комп'ютерна програма для автоматизації розрахунку параметрів.

Далі проводиться серія експериментів, і на підставі отриманих даних, з використанням обраного методу ідентифікації, розраховуються значення параметрів моделі. Для знайдених значень параметрів за тими же експериментальними даними розраховуються оцінки адекватності моделі. Значення цих оцінок мають бути значно кращими, ніж поставлені в постановці задачі моделювання вимоги до адекватності моделі, інакше результати ідентифікації є незадовільними і необхідно повторити процедуру або переглянути структуру моделі.

Вдало ідентифікована модель проходить перевірку адекватності на іншій серії експериментів.

Динамічні моделі використовуються для прогнозування поведінки та дослідження змін нестационарних об'єктів у часі. Для стаціонарних об'єктів розробляються статичні моделі.

1.3.2 Особливості аналітичного статичного моделювання

Якщо за підсумками аналізу характеристик об'єкта моделювання він був віднесений до класу стаціонарних (вихід залежить тільки від входу, об'єкт не змінюється в часі), то можливе застосування даної методики, яка є окремим випадком загальної методики моделювання.

У випадку, коли на стадії аналізу вдалося встановити майже всі зв'язки між характеристиками об'єкта, це відображається в концептуальній моделі, за якою відбувається розробка формальних залежностей моделі. Зазвичай, статична модель подається у вигляді одного рівняння або системи рівнянь чи нерівностей, лінійних чи нелінійних, алгебраїчних чи трансцендентних. Якщо інформації про залежність характеристик недостатньо для їх точної ідентифікації, або зв'язки мають випадковий характер, об'єкт моделювання розглядається як «чорна скриня», стосовно якої виділяються лише входи та виходи, а внутрішні зв'язки вважаються невідомими. Для ідентифікації моделі в цьому випадку може бути застосований апарат регресійного аналізу. Розглянемо ситуацію, коли у об'єкта одна вихідна величина та декілька вхідних. Такі об'єкти в концептуальній моделі подаються у вигляді, зображеному на рис 1.2, на якому x_1, \dots, x_n – вхідні змінні (фактори, регресори), y – вихідна змінна (відклик).

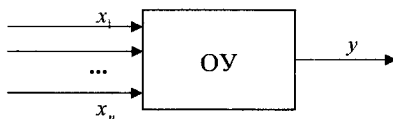


Рисунок 1.2 – Об'єкт моделювання «чорна скриня»

При цьому приймаються припущення, які визначають стаціонарність об'єкта, відсутність взаємозв'язку між різними реалізаціями процесу. Наприклад, як відклик можна розглянути витрати палива автомобілем на 100 км шляху. Факторами будуть вага вантажу, середня швидкість руху. Припущення: рух відбувається в основному по горизонталі по однотипних дорогах; якість пального, технічний стан автомобіля та водій не змінюються.

Структура математичної моделі може бути лінійною чи нелінійною. Розглянемо найпростіший випадок, коли залежність вихідної змінної від факторів має вигляд:

$$y = b_0 + b_1 \cdot x_1 + \dots + b_n \cdot x_n, \quad (1.1)$$

де b_0, \dots, b_n – невідомі параметри.

Для пошуку значень параметрів полінома (1.1) може бути застосований один з методів параметричної ідентифікації, наприклад, метод найменших квадратів (МНК, least squares method), який докладніше викладений у розділі 6.

Методика застосування МНК складається з етапів:

- збір статистичних даних про залежність відгуку від факторів;
- обробка і дослідження цих даних;
- розрахунок варіантів значень параметрів (множинний регресійний аналіз) і їх оцінок адекватності.

Збір статистичних даних здійснюється в ході експерименту. Результати експерименту подаються у вигляді таблиці 1.2.

Таблиця 1.2 – Результати експерименту

| № | y | x_1 | ... | x_n |
|-----|-------|----------|-----|----------|
| 1 | y_1 | x_{11} | ... | x_{1n} |
| ... | ... | ... | ... | ... |
| m | y_m | x_{m1} | ... | x_{mn} |

Рядки в таблиці відповідають результатам окремого експерименту. Кількість записів у таблиці має дорівнювати або бути більше числа факторів. Чим більше експериментальних даних, тим більше інформації для побудови моделі. Однак слід пам'ятати, що кожний експеримент має певну вартість.

З метою дослідження об'єкта дані експерименту піддаються статистичній обробці та аналізу. На основі даних:

- розраховуються середнє значення, середньоквадратичне відхилення, мінімум, максимум кожного з факторів і відгуку;
- перевіряються гіпотези про закон розподілу факторів і відгуку шляхом побудови гістограм;
- обчислюється кореляційна матриця.

Дослідження статистичної інформації дає змогу частково виявляти помилки введення та методики проведення експерименту. Аналіз кореляційної матриці дозволяє з'ясувати ступінь взаємозв'язку відгуку з факторами та факторів між собою.

Для проведення розрахунків результати експерименту подаються у вигляді матриць:

$$U = \begin{pmatrix} 1 & x_{11} & \dots & x_{1n} \\ \dots & \dots & \dots & \dots \\ 1 & x_{m1} & \dots & x_{mn} \end{pmatrix}, \quad Y = \begin{pmatrix} y_1 \\ \dots \\ y_m \end{pmatrix}$$

Розрахунок значень стовпця параметрів B полінома (1.1) відбувається за формулою МНК:

$$B = [U^T \cdot U]^{-1} \cdot U^T \cdot Y.$$

Отриманий поліном визначає середньостатистичне значення відгуку при заданих значеннях факторів. Це означає, що для адекватної моделі значення її вихідної змінної може значно відрізнятись від значення вихідної величини об'єкта, але на великій кількості спостережень при однакових значеннях факторів середнє експериментальне значення відгуку буде близьким до розрахованого за моделлю.

Для оцінювання якості побудованої моделі використовують різні статистичні оцінки (наприклад, залишкову дисперсію, критерій Фішера та інші). З метою поліпшення цих оцінок можна застосовувати такі прийоми:

- видалення з моделі малозначущих змінних;
- перегляд експериментальних даних та видалення ізольованих точок;
- внесення до структури моделі нелінійних складових.

Для видалення з моделі малозначущих змінних застосовують процедуру багатокрокового регресійного аналізу. Його суть полягає в тому, що після знаходження для полінома (1.1) значень параметрів та оцінок якості для кожного фактора розраховують статистичну значимість стосовно впливу на відгук. Зі структури моделі та з інформаційної таблиці видаляється фактор з найменшою значимістю і його експериментальні дані. Після цього процедура розрахунку значень параметрів і оцінок якості моделі повторюється. Якщо без цього фактора оцінки моделі покращилися, то його видалення вважається успішним і переходять до наступного фактора з найменшою значимістю. Інакше процедура припиняється і оптимальним за структурою є останній успішно знайдений поліном.

Поліпшення оцінок полінома також можна досягти шляхом видалення частини рядків з інформаційної матриці. Видаляються рядки, при підстановці з яких значень факторів відповідне значення відгуку моделі дуже сильно, порівняно з іншими рядками, відрізняється від фактичного значення вихідної величини об'єкта (так звані ізольовані точки). Обґрунтованість видалення полягає в тому, що, можливо, ці дані не є типовими для функціонування об'єкта. При кожному видаленні необхідно приймати рішення, наскільки дані в рядку типові для об'єкта, і

контролювати зміну значень оцінок моделі. Видалити можна лише невелику частину даних, щоб не погіршити їх статистичну достовірність в цілому.

Якщо лінійна структура моделі не дозволяє досягти необхідного рівня значень оцінок критеріїв якості, до неї можна додавати нелінійні компоненти (наприклад, добуток факторів чи їх ступені). Для застосування МНК нелінійні складові перетворюють у лінійні шляхом спеціальних заміни, перераховують інформаційну матрицю, а після розрахунку значень параметрів лінеаризованої моделі оберненою процедурою визначають параметри нелінійної моделі. Наприклад, поліном $y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_1^2$ може бути легко перетворений на лінійний $y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2$ шляхом заміни $x_2 = x_1^2$, причому параметри полінома залишаються ті самі.

Для розробленої моделі завжди виконується перевірка адекватності на основі нової серії експериментів.

Як видно з зазначеного вище, розробка аналітичних моделей є кропіткою та витратною справою. Крім того, однією з проблем аналітичного моделювання є врахування істотних випадкових впливів на об'єкт. З цією задачею легше впоратися, використовуючи арсенал імітаційного моделювання, який спирається на засоби ЕОМ.

1.4 Комп'ютерне та імітаційне моделювання

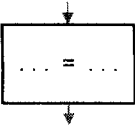
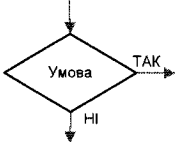
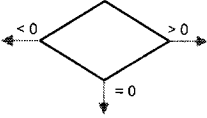
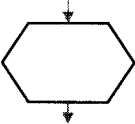
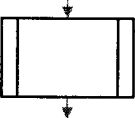
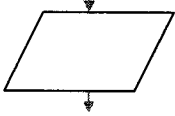

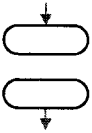
1.4.1 Алгоритми і програмування

Термін «алгоритм» («алгорифм», «алгоризм») походить від імені узбецького математика Аль-Хорезмі, який в IX сторіччі розробив правила арифметичних дій над числами в десятковій системі.

Алгоритм (algorithm) – це правило, яке сформульовано певною мовою та визначає процес переробки допустимих вхідних даних в результати пошуку. Алгоритм характеризується: детермінованістю (визначеністю) – однозначністю результату процесу при заданих вхідних даних; дискретністю – розподіленням алгоритмічного процесу на окремі елементарні акти, можливість виконання яких людиною або машиною не викликає сумнівів; масовістю – вихідні дані для алгоритму можна вибирати з певної безлічі даних (потенційно безкінечних); зрозумілістю для виконавця.

У структурі алгоритмів можна виділити ряд блоків, які є сукупністю елементарних операцій, що виконують певну функцію. Типовими блоками в структурній схемі алгоритму є: процес, рішення, модифікація, попередньо визначений процес, введення-виведення, з'єднувач, пуск-зупинка. Їх умовні позначення наведені у табл. 1.3.

Таблиця 1.3 – Типові блоки у структурній схемі алгоритму

| | |
|---|--|
|  | <p>Процес – обчислювальний блок</p> |
|  | <p>Арифметичний вираз</p> |
|  | <p>Розв'язок – умовний блок</p> |
|  | <p>Модифікація – зміна команди або програми</p> |
|  | <p>Наперед визначений процес – використання готових алгоритмів або програм</p> |
|  | <p>Введення – виведення даних</p> |
|  | <p>З'єднувач – вказівник зв'язку між перерваними схемами</p> |
|  | <p>Пуск-зупинка – початок або кінець алгоритму</p> |

Алгоритм може мати лінійну, розгалужену та циклічну структури. Лінійна структура характеризується відсутністю умовних блоків. На рис.1.3 наведений приклад алгоритму для вирішення завдання зміни місць вмісту двох елементів пам'яті комп'ютера R і P . Причому, виявляється необхідним використання третьої комірки C .

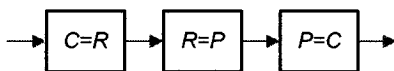


Рисунок 1.3 – Приклад алгоритму

Як приклад розгалуженої структури алгоритму, яка має хоч один умовний блок, може слугувати алгоритм вибору найбільшого значення змінних N і M (рис. 1.4).

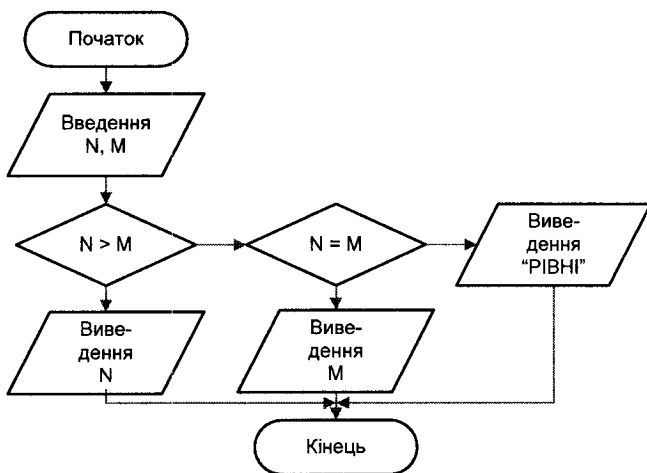


Рисунок 1.4 – Розгалужена структура алгоритму

Широко розповсюдженою при вирішенні інженерних завдань є циклічна структура (рис. 1.5), де 1 – підготовка до першого виконання тіла циклу; 2 – тіло циклу, яким називається частина обчислювального процесу, що багато разів повторюється; 3 – підготовка до чергового виконання тіла циклу; 4 – управління циклом, яке виконує перевірку на закінчення циклу.

Розрізняють такі види циклів:

- із заданим або обчислюваним числом повторень;
- ітераційні, в яких число повторень заздалегідь невідоме;
- складні, з розгалуженням у тілі циклу і вкладеними циклами.

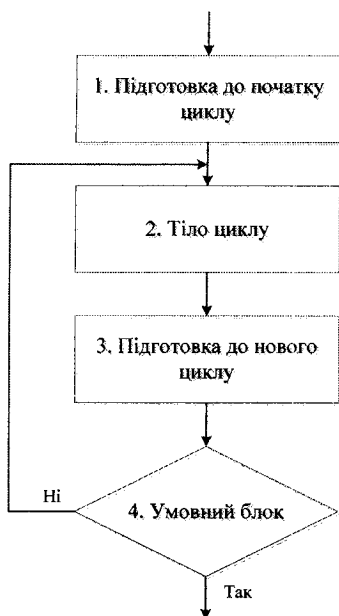


Рисунок 1.5 – Циклічна структура алгоритму

Також існують інші способи запису алгоритмів – у вигляді операторної або граф-схеми.

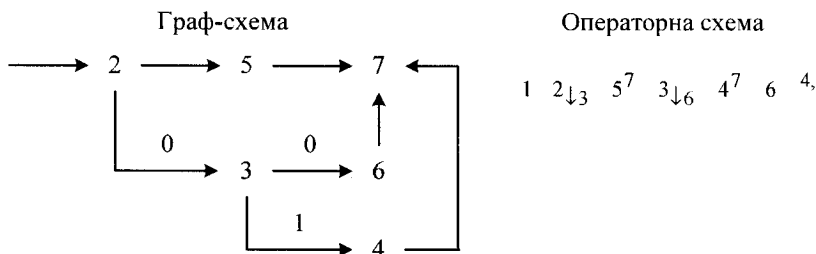
Для зображення операторної схеми алгоритму використовуються арифметичні і логічні оператори. Арифметичні оператори виконують дії, пов'язані з обчисленнями.

Позначимо операторів великими буквами латинського алфавіту з індексами, які вказують номер оператора. Після виконання операцій, передбачених арифметичним оператором, процес обчислень може бути продовжений єдиним шляхом, незалежно від результатів, що видаються оператором. Передача управління від арифметичного оператора позначається номером того оператора, якому передається управління, та записується справа вгорі від символу даного оператора. Наприклад, запис A_p^S означає, що від оператора управління A_p передається операторові з номером S .

Логічні оператори призначені для перевірки виконання заданих умов. Позначимо їх буквою P з вказанням номера оператора. Після реалізації логічного оператора управління передається одному з двох операторів залежно від виконання умови, що перевіряється. Передача управління від логічного оператора позначається стрілками з номерами тих операторів,

яким передається управління. Наприклад, $P_{K\downarrow j}^{\uparrow i}$ означає, що від логічного оператора P_K керування передається операторові з номером i , якщо умова, яка перевіряється оператором, виконана, або ж оператору з номером j , якщо вона не виконана. Для операторів, як арифметичних, так і логічних, позначення передачі від одного оператора до іншого, наступного за ним, опускається.

Передача управління даному оператору від інших позначається номером того оператора, від якого передається управління, та записується зліва зверху від символу даного оператора. Наприклад, запис $l, n A_m$ означає, що оператору A_m управління передається від оператора з номерами l і n . В цьому випадку алгоритм, структура якого наведена на рис. 1.4, може бути записаний таким чином (блоки позначаються цифрами):



У наш час велику популярність набули методи структурного програмування (structured programming), де як єдині три види структур алгоритмів застосовуються ЗЧЛЕНУВАННЯ, ВИБІР і ПОВТОРЕННЯ. Основним методом створення програм при цьому є алгоритм покрокової деталізації, в якому без складання блок-схеми програміст поступово пересувається текстом програми, послідовно організовуючи і деталізуючи шари, які відповідають різним рівням абстракції, використовуючи при цьому спеціальну універсальну мову структурного програмування, наприклад PDL. Складена таким чином програма може бути легко і однозначно перекладена на будь-яку, зручну для користувача, мову програмування.

Знаходження алгоритмів вирішення різних класів завдань – одна з цілей математики. Мета ж прикладної математики стосовно використання комп'ютерів – знаходження комп'ютерних алгоритмів розв'язання практичних (інженерних) завдань.

1.4.2 Особливості комп'ютерного моделювання

Багато явищ і процесів різної природи описуються аналогічними співвідношеннями, наприклад, електроакустична аналогія, електро-, магніто- і гідродинаміка. Тому для аналізу (розв'язання, розрахунку) математичних моделей необхідно володіти розвиненим математичним апаратом, що охоплює всі види типових завдань прикладної математики. Стосовно використання комп'ютерів основним етапом розрахунку математичних моделей є їх алгоритмізація, тобто розробка структури алгоритму, поданого у вигляді блок-схеми, граф-схеми або програмно-реалізованого з використанням принципів структурного програмування.

Комп'ютерне моделювання – метод розв'язування задачі аналізу або синтезу складної системи, що ґрунтується на використанні її комп'ютерної моделі. Сутність комп'ютерного моделювання полягає у відшукуванні кількісних і якісних результатів із залученням наявної моделі.

Комп'ютерна модель складної системи має якомога повніше відбивати всі основні фактори й взаємозв'язки, що характеризують реальні ситуації, критерії та обмеження. До того ж модель має бути настільки універсальною (щоб охоплювати якнайширше коло близьких за призначенням об'єктів) настільки й простою (щоб сприяти виконанню необхідних досліджень із мінімальними витратами).

Комп'ютерний напрям моделювання в науці отримав назву обчислювального експерименту.

Обчислювальний експеримент (computational experiment) – це методологія дослідження, заснована на вивченні математичної (інформаційної) моделі за допомогою логіко-математичних алгоритмів на комп'ютері.

Комп'ютерне моделювання (обчислювальний експеримент) має істотні переваги перед натурним експериментом.

По-перше, не потрібно проводити експеримент на реальних фізичних, економічних чи інших об'єктах, тому затрати на різні комп'ютерні експерименти набагато менші, ніж на натурні експерименти. Масштаби експериментів можна вибрати на свій розсуд, при цьому є можливість проведення багатократних дослідів із поступовими змінами вхідних даних задачі.

По-друге, проведення реальних експериментів у деяких галузях науки небезпечне (екологія, ядерна фізика) або неможливе (астрофізика).

По-третє, у процесі побудови математичних моделей для проведення обчислювального експерименту і під час їхнього дослідження можна проаналізувати і зрозуміти характеристики досліджуваного об'єкта.

1.4.3 Особливості імітаційного статистичного моделювання

Аналітичне імовірнісне моделювання полягає в побудові моделей, які оперують не з конкретними випадковими числовими послідовностями, а безпосередньо з їх імовірнісними (закони розподілу ймовірності) і спектральними (спектральні щільності або кореляційні функції) характеристиками. У загальному випадку побудова аналітичних імовірнісних моделей є надто складною обчислювальною задачею, що не дозволяє повною мірою використовувати такі їх переваги, як можливість точного аналітичного задання характеристик випадкових процесів, відсутність необхідності генерації і обробки великих вибірок випадкових чисел, пристосованість до оперативної оптимізації. У літературі описані результати досліджень, направлених на створення спеціалізованих проблемно-орієнтованих систем і пакетів прикладних програм, що об'єднують чисельні алгоритми розв'язання найбільш характерних обчислювальних процедур аналітичного імовірнісного моделювання і способи описання структур систем. Ці засоби мають, як правило, вузькоспеціалізовану спрямованість на моделювання певних класів складних систем. Тому, основним методом стохастичного комп'ютерного моделювання є метод статистичного імітаційного моделювання.

Методика статистичного моделювання охоплює ряд послідовних етапів: моделювання псевдовипадкових числових послідовностей з заданою кореляцією і законом розподілу ймовірностей, що імітують вхідні сигнали і збурення в системі; моделювання перетворення отриманих числових послідовностей в системі; статистична обробка результатів моделювання. Розглянемо перераховані етапи.

Комп'ютерне моделювання псевдовипадкових числових послідовностей з заданими характеристиками. При побудові імітаційної моделі системи виникає завдання отримання псевдовипадкових числових послідовностей з заданою кореляцією і законом розподілу ймовірностей. Відомим є метод отримання числових послідовностей з заданими статистичними характеристиками за допомогою сортування вихідних послідовностей. Метод заснований на тому, що коефіцієнт кореляції випадкових чисел більше залежить від порядку їх розташування в числовому ряді, ніж від величини.

У випадку з методом сортування генерується псевдовипадкова послідовність $X(n)$ із заданою кореляційною функцією, але довільним розподілом. У відповідність їй ставиться послідовність цілих чисел $I(n) = n$. Потім обидві послідовності попарно сортуються. При цьому величини $X(n)$ розміщуються за зростанням, а масив $I(n)$ запам'ятовує їх попереднє положення (місця в неупорядкованому масиві $X(n)$). Таким чином, цілочисловий масив $I(n)$ відображає кореляцію між елементами масиву $X(n)$. Після упорядкування масив $X(n)$ не викликає інтересу, оскільки вся

інформація про кореляційну функцію тепер міститься в масиві $I(n)$. Потім генерується псевдовипадкова послідовність $Y(n)$ із заданим розподілом і нульовою кореляцією і записується на місце масиву $X(n)$. Після цього вона сортується у порядку зростання. Далі масиви $I(n)$ і $Y(n)$ попарно сортуються, причому масив $I(n)$ розташовується у порядку зростання. Структурна схема алгоритму наведена на рис. 1.6, а. У результаті виконання даного алгоритму отримуємо псевдовипадкову числову послідовність, яка містить величини, що розподілені за заданим законом і мають задану кореляційну функцію. Алгоритм сортування доцільно використовувати у випадках, коли для статистичного моделювання системи є невеликий обсяг статистичних даних, що не вимагають операції над числовими масивами великої розмірності.

Відомим є алгоритм фільтрації, що вимагає менших витрат машинного часу для здобуття випадкового процесу з заданою кореляцією і законом розподілу ймовірностей. Як вихідний вибирається нормальний стаціонарний випадковий процес $X(t)$. Завжди існує таке нелінійне неінверсійне перетворення $Y = W_N(X)$, яке перетворює нормальну функцію щільності $f_X(x)$ процесу $X(t)$ в задану функцію щільності $f_Y(y)$. Якщо вихідний процес $X(t)$ має кореляційну функцію $R_{XX}(\tau)$, то перетворений процес $Y(t)$ матиме кореляційну функцію $R_{YY}(\tau)$, що відрізняється від функції $R_{XX}(\tau)$ і пов'язана з нею деякою залежністю $R_{YY} = \varphi(R_{XX})$. Вигляд цієї залежності визначається перетворенням $Y = W_N(X)$. Для того, щоб кореляційна функція перетвореного процесу була потрібною, необхідно вибрати кореляційну функцію вихідного процесу:

$$R_{XX}(\tau) = \varphi^*[R_{YY}(\tau)],$$

де φ^* – функція, обернена до φ .

При використанні даного способу підготовча робота складається з декількох етапів:

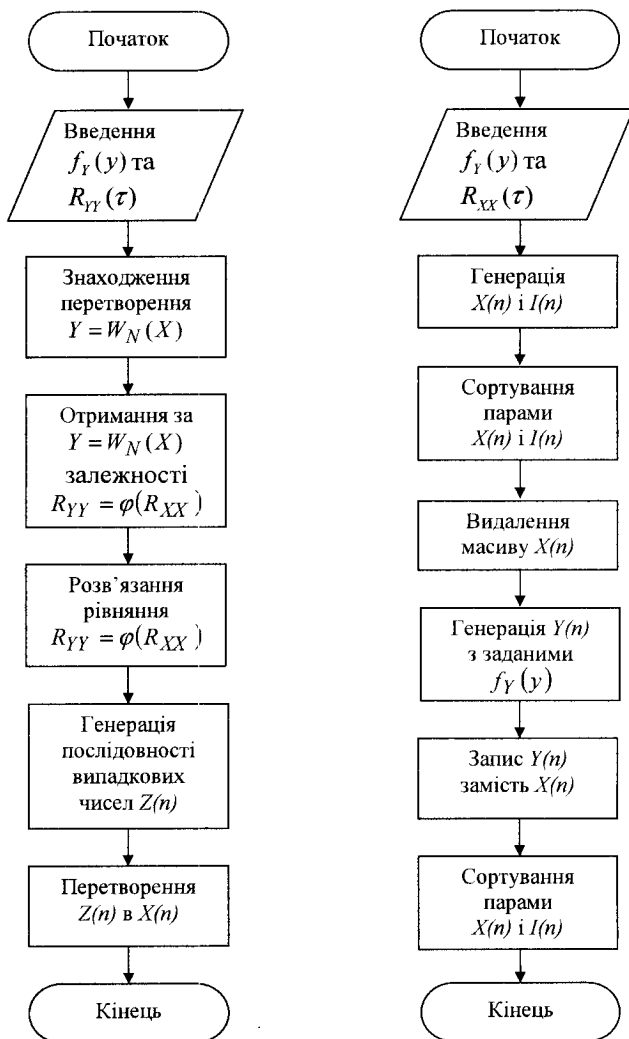
- знаходження функції перетворення $Y = W_N(X)$ за заданою функцією щільності $f_Y(y)$;
- отримання за знайденою функцією $Y = W_N(X)$ залежності $R_{YY} = \varphi(R_{XX})$;
- розв'язання рівняння $R_{YY} = \varphi(R_{XX})$ відносно R_{XX} , тобто визначення кореляційної функції $R_{XX}(\tau)$ вихідного нормального процесу $X(t)$.

Після закінчення підготовчої роботи моделювання випадкового процесу з заданими характеристиками зводиться до формування

дискретних реалізацій $X(n)$ нормального випадкового процесу $X(t)$ і до перетворення цих реалізацій за формулою

$$Y(n) = W_N \{X(n)\}.$$

Структурна схема описаного алгоритму наведена на рис. 1.6, б.



а)

б)

Рисунок 1.6 – Структурні схеми алгоритмів

Описаний алгоритм вимагає менших витрат машинного часу, ніж алгоритм сортування, не вимагає накопичення і зберігання в пам'яті числових масивів великої розмірності. Недоліками методу є його складність для програмування і великий обсяг підготовчої роботи. Ускладнення принципового характеру полягає в тому, що в загальному випадку не є можливим довести існування розв'язку рівняння $R_{YY}(\tau) = \varphi[R_{XX}(\tau)]$ відносно $R_{XX}(\tau)$.

При використанні обох описаних алгоритмів виникає задача генерування на комп'ютері псевдовипадкових числових послідовностей з заданими законами розподілу і нульовою кореляцією, та псевдовипадкових числових послідовностей з заданою кореляційною функцією і довільним розподілом. Задачу генерування випадкових чисел із заданим законом розподілу вирішують у декілька етапів. Спочатку отримують послідовність рівномірно розподілених на інтервалі $[0, 1]$ псевдовипадкових чисел, а з неї – послідовність псевдовипадкових чисел із заданим законом розподілу.

Розглянемо алгоритмічні (програмні) методи отримання рівномірно розподілених випадкових чисел (на практиці інколи застосовують також фізичне моделювання з використанням спеціальної комп'ютерної приставки). Суть програмних методів полягає в тому, що рівномірно розподілені псевдовипадкові числа обчислюють за допомогою деякої рекурентної формули, де кожне наступне $(i+1)$ -ше значення утворюється з попереднього (групи попередніх) i шляхом застосування деякого алгоритму, що містить логічні та арифметичні операції.

Існує велика кількість способів імітації рівномірного розподілу (методи формування залишків, поетапного підсумовування, зрізання, перемішування). Загальними для всіх цих методів є вимоги, що застосовуються до генерованої послідовності рівномірно розподілених випадкових чисел: кількість операцій для отримання кожного псевдовипадкового числа має бути мінімальною; випадкові числа генеруються якомога менш корельованими, а їх розподіл – близьким до рівномірного, причому вигляд розподілу і ступінь корельованості чисел не повинні змінюватися у процесі роботи програми.

У стандартному математичному і програмному забезпеченні різних типів комп'ютерів є спеціальні процедури і підпрограми для генерації рівномірно розподілених послідовностей псевдовипадкових чисел.

Використовуючи рівномірно розподілену на інтервалі $[0, 1]$ випадкову величину X , можна отримати послідовність випадкових чисел з довільно заданим законом розподілу ймовірностей. Розрізняють три основні способи формування таких послідовностей:

1) пряме перетворення числа X_i , що є реалізацією випадкової X величини, рівномірно розподіленої на інтервалі $[0, 1]$ за допомогою

деякої функції W_N , в число Y_i , яке може розглядатися як реалізація випадкової величини Y , що має заданий закон розподілу;

2) відсіювання чисел з первинної послідовності рівномірно розподілених на інтервалі $[0, 1]$ псевдовипадкових чисел так, щоб числа, що залишилися, були розподілені за заданим законом;

3) моделювання умов відповідних граничних теорем теорії ймовірностей.

Широкого поширення набувають способи прискореної генерації випадкових чисел. Так, значний ефект збільшення швидкодії при імітації випадкових чисел, розподілених за нормальним законом, у порівнянні зі способом, заснованим на використанні центральної граничної теореми теорії ймовірностей, дає спосіб, що містить запропонований Мюллером алгоритм, за яким пара незалежних рівномірно розподілених на відрізку $[0,1]$ чисел x_1, x_2 перетворюється в пару незалежних нормально розподілених величин

$$\xi_1 = \sqrt{-2 \ln x_2} \cos 2\pi x_1; \xi_2 = \sqrt{-2 \ln x_1} \cos 2\pi x_2.$$

Відзначимо, що цей спосіб є теоретично точним і вимагає найменшої кількості рівномірно розподілених чисел $X(n)$.

Відомі спеціальні способи здобуття випадкових чисел, що підпорядковуються ряду законів розподілу вірогідності. Наприклад, розподіл Релея визначається одним параметром, рівним середньому квадратичному відхиленню вихідного двовимірного нормального розподілу. Звідси витікає такий спосіб імітації розподілу Релея:

$$\eta_i = \sigma \sqrt{\xi_1^2 + \xi_2^2},$$

де η_i – випадкове число, розподілене за законом Релея, а ξ_1 та ξ_2 – випадкові числа, що мають нормальний розподіл з нульовим математичним сподіванням і рівним одиниці середнім квадратичним відхиленням.

Існує також співвідношення, що пов'язує випадкові числа, розподілені згідно з законом Релея, з випадковими числами x_i , рівномірно розподіленими на відрізку $[0, 1]$, яке визначає інший спосіб генерації:

$$\eta_i = \sigma \sqrt{-2 \ln x_i}.$$

Для імітації закону розподілу Максвелла можна скористатися тим, що випадкова величина, яка підлягає розподілу Максвелла, може розглядатися як модуль тривимірного випадкового вектора, проекції якого на осі координат підлягають нормальному розподілу з рівними між собою середнім квадратичним відхиленням і математичними сподіваннями, що

дорівнюють нулю. Тому можна скористатися нижченаведеною формулою для імітації закону розподілу Максвелла:

$$\theta_i = \sigma \sqrt{\xi_1^2 + \xi_2^2 + \xi_3^2},$$

де θ_i – випадкове число, розподілене за законом Максвелла; ξ_1, ξ_2, ξ_3 – випадкові числа, що мають нормальний розподіл з математичними сподіваннями, рівними нулю, і середніми квадратичними відхиленнями, рівними одиниці.

Розглянемо способи здобуття псевдовипадкових числових послідовностей з заданою кореляційною функцією. Відомий алгоритм, де n некорельованих випадкових чисел x_i, \dots, x_n піддають такому лінійному перетворенню, після якого отримані величини y_i, \dots, y_n мають задану кореляційну матрицю $K(R_{nm})$.

Таким чином, величини знаходяться з матричного рівняння

$$K(Y) = W_N [K(X)],$$

де W_N – лінійне перетворення вектора-стовпця $K(X)$ в $K(Y)$.

У розгорнутому вигляді отримуємо:

$$\begin{aligned} y_1 &= a_{11}(x_1 - m_x) + m_{y1}; \\ y_2 &= a_{12}(x_1 - m_x) + a_{22}(x_2 - m_x) + m_{y2}; \\ &\dots\dots\dots \\ y_n &= a_{1n}(x_1 - m_x) + a_{nn}(x_n - m_x) + m_{yn}, \end{aligned}$$

де коефіцієнти перетворення a_{ik} визначаються з рівняння вигляду

$$R_{IK} = a_{1l}a_{lk} + a_{2l}a_{2k} + \dots + a_{ll}a_{lk}$$

а $K[m_Y]$ – вектор-стовпець математичних сподівань y_i .

При великих значеннях n викладений спосіб генерування корельованих псевдовипадкових послідовностей стає незручним для реалізації на ЕОМ, оскільки запам'ятовування елементів матриці $K(a)$ вимагає великого об'єму оперативної пам'яті ($N = n(n+1)/2$ комірок) і великого обсягу обчислень (витрат машинного часу). У зв'язку з цим у ряді випадків виявляється зручнішим моделювання корельованих випадкових процесів за методом канонічних розкладів. Нехай неперервний випадковий процес $Y(t)$ заданий канонічним розкладом

$$Y(t) = \sum_{k=1}^{\infty} V_k J_k(t), \quad (1.2)$$

де V_k – некорельовані випадкові коефіцієнти; $J_k(t)$ – система деяких детермінованих координатних функцій.

Цифрове моделювання (digital simulation) випадкового процесу, заданого канонічним розкладом, здійснюється таким чином. Як V_k використовуються значення некорельованих випадкових величин X_k , а нескінченний ряд (1.2) при обчисленнях приблизно замінюється зрізаним кінцевим рядом. Використовуючи канонічний розклад, отримуємо співвідношення для моделювання

$$Y_n = \sum_{k=1}^N V_k J_k(n),$$

у якому дисперсії D_{V_k} некорельованих випадкових величин x_k і дискретні координатні функції $J_k(n)$ знаходяться з таких рекурентних співвідношень:

$$D_{V_k} = R_{YY}(1, 1);$$

$$J_1(n) = \frac{1}{D_{V_1}} R_{YY}(n, 1);$$

$$D_{V_k} = R_{YY}(k, k) - \sum_{i=1}^{k-1} D_{V_i} J_i(k), \quad k = \overline{2, N};$$

$$J_k(n) = \frac{1}{D_{V_k}} \left[R_{YY}(n, k) - \sum_{i=1}^{k-1} D_{V_i} J_i(n) J_i(k) \right].$$

Звідси отримуємо

$$Y_n = \sum_{k=1}^N \sigma_k x_k J_k(n).$$

Послідовність Y_n матиме нормальний розподіл і задану кореляційну функцію $R_{XX}(\tau)$.

Методи сортування і фільтрації у поєднанні з розглянутими способами генерування псевдовипадкових числових послідовностей з заданими кореляційними функціями дозволяють отримати послідовності випадкових чисел, що імітують вхідні сигнали і збурювальні дії в об'єкті моделювання.

Моделювання процесів перетворення випадкових сигналів.

Розглянемо способи побудови алгоритмів для моделювання процесів перетворення випадкових сигналів різними перетворювачами і системами. При моделюванні перетворення випадкових сигналів лінійними динамічними системами зручно користуватися їх імпульсною характеристикою $g(t)$.

Для отримання цифрової моделі перетворення вхідного випадкового сигналу необхідно моделювати операцію згортки функцій $x(t)$ та $g(t)$.

Здійснюючи заміну інтеграла Дюамеля сумою дискретних значень за методом прямокутників, отримуємо

$$y(n) = \sum_{k=0}^{n-m} g(k)x(n-k), \quad (1.3)$$

де m – дискретний аналог тривалості перехідного процесу.

Існують також точніші методи інтегрування – квадратурні формули Ньютона-Котеса (трапецій, Сімпсона тощо). У цих випадках формула (1.3) набуває вигляду

$$y(n) = \sum_{k=0}^n c(k)g(k)x(n-k),$$

де коефіцієнти $c(k)$ визначаються залежно від методу чисельної інтеграції, що був застосований.

Істотною перевагою методу моделювання лінійних динамічних систем на основі дискретного аналога інтеграла згортки є можливість узагальнення його на випадок моделювання лінійних динамічних систем зі змінними параметрами (нестационарних систем).

Нестационарна система (non-stationary system) описується імпульсною характеристикою $g(t, \tau)$, залежною від двох змінних. У цьому випадку реакцію нестационарної системи на випадковий сигнал запишемо у вигляді

$$y(t) = \int_0^t x(\tau)g(\tau, t-\tau)d\tau,$$

або в дискретній формі

$$y(n) = \sum_{k=0}^{n-1} x(k)g(k, n-k)c(n-k).$$

Описаний метод застосовується у тих випадках, коли моделювання вхідного випадкового сигналу здійснюється за допомогою алгоритмів нелінійного перетворення, фільтрації і моделювання умов граничних теорем теорії вірогідності.

Відзначимо, що у разі моделювання вхідного випадкового сигналу (random signal) на основі алгоритму сортування (див. рис. 1.6, а) побудувати імітаційну модель системи неможливо і досліджувати її доцільно матричними методами.

Структурна схема алгоритму моделювання наведена на рис. 1.7.



Рисунок 1.7 – Структурна схема алгоритму моделювання нестационарної системи

Цифрові моделі замкнутих нелінійних систем, що є комбінаціями лінійних динамічних і нелінійних статичних перетворювачів, можна побудувати на основі алгоритмів, за якими моделюють перетворення сигналів в окремих перетворювачах (рис. 1.8).

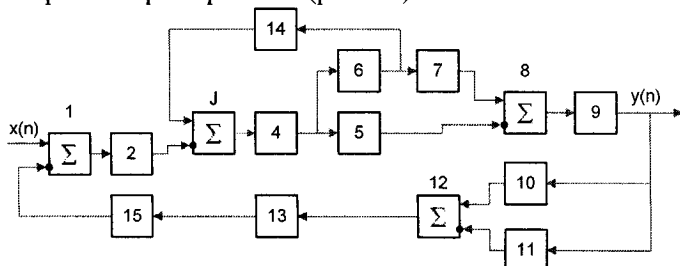


Рисунок 1.8 – Цифрова модель замкнутої нелінійної системи

Моделювання описаних систем часто пов'язане зі значними труднощами, проте у ряді випадків структурну схему системи можна представити у спрощеному вигляді (рис. 1.9).

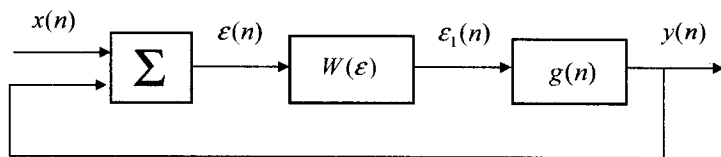


Рисунок 1.9 – Спрощений вигляд структурної схеми

У цьому випадку

$$\begin{aligned} \varepsilon(n) &= x(n) - y(n); \\ \varepsilon_1(n) &= W_N[x(n) - y(n)], \end{aligned} \quad (1.4)$$

де W_N – оператор нелінійного перетворювача системи.

Застосовуючи далі описаний вище алгоритм для цифрового моделювання лінійних динамічних систем, отримуємо

$$y(n) = \sum_{k=0}^{n-1} g(k) \varepsilon_1(n-k).$$

Таким чином, особливістю цифрових моделей нелінійних замкнених систем є необхідність розв'язання на кожному кроці нелінійних

алгебраїчних рівнянь вигляду (1.4) за умови, що лінійні динамічні ланки системи моделюються на основі дискретної згортки.

Рішення цієї задачі можна спростити, якщо в ланцюг зворотного зв'язку системи ввести елемент затримки на один період. Тоді нелінійне рівняння (1.4) перетвориться в рекурентне:

$$y(n) = \sum_{k=0}^{n-1} g(k)W_N \{x(n-k-1) - y(n-k-1)\}.$$

Введення в ланцюг зворотного зв'язку елемента затримки вносить додаткову похибку до цифрової моделі. Проте на кроці дискретизації еквівалентна дискретна система з елементом затримки так само, як і без нього, збігається з вихідною неперервною системою. Тому вибором кроку дискретизації можна досягти як завгодно малого впливу похибки затримки.

Завершальним етапом статистичного моделювання є статистична обробка результатів.

Кількість реалізацій і точність обчислень. Кількість реалізацій при вирішенні завдань методом статистичного моделювання визначається необхідною точністю отримуваних результатів.

Нехай метою моделювання буде обчислення ймовірності P появи деякої випадкової події A (наприклад, при дослідженні точності контролю розмірів виробів, ймовірності належності розмірів виробу до певної сортувальної групи; ймовірності того, що виріб буде застрахований при вторинному контролі, якщо при первинному контролі його визнано придатним). За оцінку для шуканої ймовірності P застосовується частота L/N настання події A при N реалізаціях, де L – кількість випробувань, при яких відбувається подія A . Враховуючи центральну граничну теорему теорії ймовірностей, частота L/N при чималих N має нормальний розподіл, який визначається математичним сподіванням $M(L/N) = P$ і дисперсією $D(L/N) = P(1-P)/N$.

Отже,

$$P(L/N < x) \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi D(L/N)}} \exp\left\{-\frac{[(L/N) - M(L/N)]^2}{2D(L/N)}\right\} d(L/N).$$

При чималих N отримуємо

$$P\left[\frac{(L/N) - M(L/N)}{\sqrt{D(L/N)}} < \frac{\varepsilon}{\sqrt{D(L/N)}}\right] \approx \Phi\left[\frac{N\varepsilon}{\sqrt{D(L/N)}}\right], \quad (1.5)$$

де $\Phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-z^2} dz$; ε – точність нерівності.

Враховуючи певну ймовірність P , знайдемо за таблицями нормального розподілу значення $D(L/N)$, що задовольняє рівняння $\Phi(t) = 0$, де $t = \varepsilon / D(L/N)$.

Надійну оцінку L/N отримуємо у вигляді

$$P[|(L/N) - P| < \varepsilon] = t_p \sqrt{P(1-P)/N}. \quad (1.6)$$

Згідно з формулою (1.6) з ймовірністю більшою 0,997 величина L/N задовольняє умову

$$|(L/N) - P| < 3\sqrt{P(1-P)/N}.$$

Таким чином, похибка методу статистичного моделювання при обчисленні ймовірності події A ніколи не перевищує величини $\varepsilon = 3\sqrt{P(1-P)/N}$ і спадає зі збільшенням числа випробувань обернено пропорційно кореню квадратному з N . Звідси можна визначити кількість реалізацій N , необхідних для отримання оцінки L/N з точністю ε і достовірністю P

$$N = t_p^2 P(1-P) / \varepsilon^2,$$

або для $P = 0,997$

$$N = 9P(1-P) / \varepsilon^2.$$

Аналогічно можна оцінити кількість реалізацій, необхідних для оцінювання за результатами моделювання середнього значення випадкової величини. Припустимо, що здійснюється формування N реалізацій випадкової величини X , що має середнє значення M і дисперсію σ^2 . Визначимо

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i.$$

В силу центральної граничної теореми теорії ймовірностей

$$P\left(\frac{\bar{X} - M}{\sigma\sqrt{N}} < \sqrt{N}/\sigma\right) \approx \Phi(\varepsilon\sqrt{N}/\sigma).$$

Тоді

$$P[|X - M| < \varepsilon] = T_p \sigma / \sqrt{N},$$

а точність

$$\varepsilon = T_p \sigma / \sqrt{N} \quad (1.7)$$

Розв'язуючи (1.7) відносно N , отримаємо

$$N = t_p^2 \sigma^2 / \varepsilon^2 \quad (1.8)$$

При $P=0,997$ формули (1.7) і (1.8), відповідно, набувають вигляду:

$$\varepsilon = 3\sigma / \sqrt{N}, \quad N = 9\sigma^2 / \varepsilon^2.$$

Похибка методу статистичного моделювання як при обчисленні ймовірності події A , так і при оцінюванні середнього значення випадкової величини складає $\varepsilon = 1/\sqrt{N}$. Зменшення похибки ε наближеного розв'язку задачі методом імовірнісного моделювання пов'язане зі значним збільшенням числа випробувань N , відповідно, і зі збільшенням часу обчислень. Наприклад, збільшення точності на порядок веде до подовження часу розв'язання задачі у сто разів.

1.5 Похибки та властивості обчислювальних алгоритмів

Сьогодні множина методів, які використовуються для комп'ютерних обчислень, ширша, ніж множина саме обчислювальних методів чи так званих методів наближених обчислень. Але на практиці все ж таки саме ці методи стають основою для побудови комп'ютерних розв'язків математичних моделей систем та процесів. В них розглядаються такі алгоритми, які найчастіше пов'язані з розв'язанням дискретної задачі, що була задана як дискретна, чи виникла як результат дискретизації вихідної неперервної задачі. Простим прикладом дискретизації може слугувати побудова різницевої схеми, що виникає при заміні диференціальних рівнянь кінцево-різницеvими співвідношеннями. При дискретизації неперервної задачі обов'язково виникає *похибка дискретизації* (sampling error). Ця похибка є однією з головних складових в *похибці обчислювального методу*.

Іншою складовою методичної похибки є *похибка округлення* (rounding error), яка виникає завдяки скінченній довжині розрядної сітки комп'ютера, що призводить до наближеного подання дійсних чисел.

Величина цієї похибки визначається не тільки довжиною розрядної комп'ютерної сітки, а й чутливістю конкретного алгоритму до похибок округлення.

Можна виділити також *похибку зрізання* (truncation error), що виникає при заміні нескінченних рядів скінченними.

Похибки можна класифікувати за способом подання на абсолютні та відносні.

Абсолютна похибка (absolute error) визначається як модуль різниці між точним A та наближеним a значеннями числа:

$$\Delta = |a - A|.$$

Відносна похибка (relative error) визначається як:

$$\delta = \frac{\Delta}{|a|}.$$

Без доведення (доведення дуже просте і для тренування його можна зробити самостійно) наведемо головні властивості арифметичних операцій з похибками:

– *абсолютна похибка суми k* (k – будь-яке ціле число) наближених чисел не перевищує суми абсолютних похибок цих чисел

$$\Delta \left[\sum_{i=1}^k a_i \right] \leq \sum_{i=1}^k \Delta_i;$$

– *відносна похибка суми k* наближених чисел не перевищує максимальної відносної похибки одного з цих чисел

$$\delta \left[\sum_{i=1}^k a_i \right] \leq \max_{1 \leq i \leq k} \delta_i;$$

– *відносна похибка добутку k* наближених чисел не перевищує суми відносних похибок цих чисел

$$\delta \left[\prod_{i=1}^k a_i \right] \leq \sum_{i=1}^k \delta_i,$$

де $\prod_{i=1}^k a_i = a_1 * a_2 * \dots * a_k$.

Використовується також відносна похибка, де в знаменнику замість $|a|$ використовується діапазон a від максимального a_{\max} до мінімального a_{\min} значення. Тому похибку називають зведеною

$$S = \frac{\Delta}{|a_{\max} - a_{\min}|}.$$

Треба розділяти *локальні* та *глобальні* похибки (local and global errors). *Локальні* виникають на кожному конкретному кроці обчислювального алгоритму, а *глобальні* охоплюють всі похибки, що виникли як на цьому кроці обчислень, так і на попередніх (ці похибки називають *похибками наслідування* чи *похибками поширення*).

При оцінюванні глобальних похибок часто використовується поняття максимальної похибки

$$\Delta_{\max} = \max_{i=1}^K \Delta_i,$$

середньої похибки

$$\Delta = \frac{1}{K} \sum_{i=1}^K |\Delta_i|,$$

середньої квадратичної похибки

$$\Delta = \sqrt{\frac{1}{K-1} \sum_{i=1}^K \Delta_i^2}.$$

Одним із головних критеріїв якості обчислювальних алгоритмів (окрім похибки, при застосуванні наближених методів) є їх швидкодія, тобто кількість елементарних обчислювальних операцій (додавання, зсув, інверсія), яка потрібна для реалізації алгоритму.

Важливим є поняття *ітераційного алгоритму* (iterative algorithm), що визначається як багатокроковий алгоритм, в якому кожний наступний крок виконується на основі попереднього. Причому можна відокремити суто ітераційні алгоритми за допомогою ітераційних процедур, де розглядаються розв'язки в окремих точках, та багатокрокові алгоритми, в яких розв'язок в попередній точці є основою для пошуку наступної точки. Збіжність є головною властивістю суто ітераційних алгоритмів: якщо вони незбіжні, то застосувати їх взагалі неможливо.

Наведемо декілька головних властивостей обчислювальних методів та алгоритмів.

Стійкість алгоритму (stability of the algorithm) – здатність виконувати обчислення і отримувати кінцевий результат із заданою точністю при зміні параметрів алгоритму і вхідних даних в деякій області, яка називається областю стійкості.

Збіжність (convergence) – це властивість алгоритму шляхом зміни його параметрів виконувати обчислення зі скільки завгодно малою похибкою для заданого класу вхідних даних (тобто при збільшенні кількості ітерацій для алгоритмів, що збігаються, похибка буде прямувати до нуля).

Причому підвищення точності досягається зміною внутрішніх параметрів алгоритму (наприклад, максимально допустимою різницею між попереднім та наступним наближенням).

Коректність обчислювального методу (the correctness of computational method) – це властивість безперечного існування розв'язку задачі та забезпечення стійкості обчислювального алгоритму, що реалізує цей метод.

1.6 Математичне моделювання і обчислювальні методи в задачах проектування комп'ютерних систем автоматики та управління

Головним методом дослідження складних процесів та систем сьогодні став обчислювальний експеримент. Це поняття було введено академіком А. А. Самарським як метод та технологія досліджень, що оснований на побудові та аналізі комп'ютерної моделі об'єкта, який вивчається.

Перший крок – це побудова математичної моделі об'єкта за допомогою вибраного виду математичного опису: алгебраїчних, диференціальних, інтегральних рівнянь, дискретної математики, теорії множин, алгебри логіки та ін.

Другим кроком є вибір методу розв'язання цих моделей. Таким методом повинен бути метод наближених чи точних обчислень, що, в свою чергу, може бути алгоритмізований та реалізований на комп'ютері. Реалізація обчислювального експерименту може бути як у вигляді готової, вже відомої програми чи пакета програм, так і нової програми, що складена вибраною мовою програмування. Після проведення обчислень настає етап обробки, аналізу та вивчення результатів, а також використання їх для корегування обчислювального експерименту та отримання висновків щодо поведінки об'єкта досліджень. Причому деякі складові цього процесу можуть бути відсутні чи змінені. Так, наприклад, за допомогою методів і програм обробки даних можуть аналізуватися результати натурного чи напівнатурного експериментів, а висновки використовуватися для корегування їх ходу та умов проведення.

В усіх випадках основу обчислювальних експериментів складає тріада: модель – метод – алгоритм (програма).

Математичне моделювання є частиною процесу створення технічних засобів комп'ютерних систем управління, автоматики та інформаційно-вимірювальної техніки. Обчислювальні методи, що програмно реалізуються на комп'ютерах, стали основним практичним інструментом розробників комп'ютерних систем та автоматичних вимірювальних інформаційних пристроїв. Накопичений великий досвід використання обчислювальної техніки, застосування чисельних методів, створення спеціального програмного забезпечення при розв'язанні різних практичних завдань в цій області. Розглянемо основні їх типи.

1. Ідентифікація динамічних характеристик лінійних ланок при використанні різних описів сигналів на їх входах і виходах.

2. Використання методу найменших квадратів для ідентифікації передавальної характеристики для масивів даних, що описують перехідну і частотну характеристики або сигнали на входах і виходах ланки.

3. Дослідження стійкості лінійних динамічних систем на основі використання різних критеріїв. Побудова області стійкості на площині параметрів системи.

4. Аналіз якості лінійних систем автоматичного управління. Визначення оптимальних дій управління – розв'язання алгебраїчного рівняння Ріккати (неперервний і дискретний випадки), до якого зводиться завдання про оптимальний лінійний регулятор. Розв'язання рівняння Ріккати пов'язано з виконанням ряду перетворень і розв'язанням окремих завдань (складання вихідної матриці, перетворення подібності), що дозволяють привести матриці до вигляду Гессенберга і вигляду Шура; знаходження власних значень матриць.

5. Дослідження нелінійних автоматичних систем на основі наближених методів розв'язання нелінійних диференціальних рівнянь. Застосування методів гармонічної лінеаризації і кусково-лінійної апроксимації.

6. Комп'ютерне статистичне імітаційне моделювання вимірювальних пристроїв і систем, яке охоплює: генерацію псевдовипадкових числових послідовностей, що імітують вимірювані величини і фактори впливу; моделювання перетворення інформативних параметрів сигналів в досліджуваних пристроях; обробку вихідних псевдовипадкових числових послідовностей (побудова гістограм, кореляційних функцій, оцінювання за критеріями).

7. Аналітичне імовірнісне моделювання вимірювальних інформаційних пристроїв і систем на основі взаємопов'язаних імовірнісних (закони розподілу ймовірності) і енергетичних (спектральні щільності потужності) моделей.

8. Розв'язання задачі визначення розподілу значень вимірюваної величини (швидкостей потоку, звукової хвилі, температур) в замкнутій області.

9. Дослідження пристроїв автоматики методом планування експерименту.

10. Аналіз спектрів різних сигналів з застосуванням перетворення Фур'є в завданнях розпізнавання образів і цифрової обробки сигналів.

11. Аналіз і оцінювання похибок вимірювальних пристроїв і систем на основі методів частинних похідних, імовірнісного моделювання, інтервального аналізу тощо.

12. Розв'язання завдань автоматизації проектування систем автоматичного управління. У цьому напрямку накопичений великий досвід у створенні різних пакетів прикладних програм.

Перелік таких задач можна було б продовжити, але й ті, що вказані, достатньо підтверджують нерозривний зв'язок процесу проектування, аналізу та синтезу пристроїв систем автоматики та управління з комп'ютерними методами математичного моделювання, обчислювальними методами та алгоритмами для їх розрахунку та реалізації.

1.7 Інструментальні засоби моделювання

1.7.1 Моделювання з використанням математичних пакетів

Сучасний процес математичного моделювання важко реалізувати без комп'ютерів або вбудованих контролерів. Для цього використовуються різноманітні інструментальні програмні засоби та середовища (**MathCad, MatLab, Mathematica, Maple, Derive, VisSim, Genius** й інші), що суттєво спрощують моделювання. Сучасні математичні пакети можна використовувати і як звичайний калькулятор, і як засоби для спрощення виразів при розв'язанні будь-яких математичних задач, і як генератори графіки або навіть звука. Стандартними стали також засоби взаємодії з Internet шляхом генерації HTML-сторінок прямо в процесі обчислень.

Звичайно, кваліфікований користувач, який в достатній мірі володіє однією з мов програмування (C++, Java, Pascal, Fortran, Prolog та іншими), може самостійно створити окрему програму або комплекс програм, що дозволить реалізувати на ПК алгоритм його задачі. Проте такий підхід потребує, як правило, великих працевитрат на програмування, відлагодження та тестування кожної програми. Тому, для скорочення часу програмування, було створено згадані прикладні програмні пакети, сфери використання яких у значній мірі перекриваються. Для найбільш ефективного використання обчислювальної техніки необхідно правильно вибрати найкращий пакет програм на ранній стадії розв'язання прикладної задачі. Адже реальна мета полягає у вирішенні певної проблеми, а обчислення – всього лише проміжний етап на шляху до цього.

При дослідженні систем автоматичного регулювання, обчислювальних математичних задач найбільш ефективним є використання програмної

системи **Matlab** з широким класом предметно-орієнтованих бібліотек (toolbox) та інструментом візуального моделювання Simulink. У системі MatLab також існують широкі можливості для програмування. Її бібліотека C Math (компілятор MatLab) є об'єктною і містить понад 300 процедур обробки даних мовою C. Усередині пакета можна використовувати як процедури самої MatLab, так і стандартні процедури мови C, що робить цей інструмент наймогутнішою підмогою при розробці додатків (використовуючи компілятор C Math, можна вбудовувати будь-які процедури MatLab у готові додатки).

Для візуалізації моделювання система MatLab має бібліотеку Image Processing Toolbox, що забезпечує широкий спектр функцій, які підтримують візуалізацію проведених обчислень безпосередньо із середовища MatLab, збільшення та аналіз, а також можливість побудови алгоритмів обробки зображень. Систему MatLab можна використовувати для обробки зображень, сконструювавши власні алгоритми, які будуть працювати з масивами графіки як з матрицями даних. Оскільки мова MatLab оптимізована для роботи з матрицями, у результаті забезпечується простота використання, висока швидкість і економічність проведення операцій над зображеннями.

Серед інших бібліотек системи MatLab можна також відзначити System Identification Toolbox – набір інструментів для створення математичних моделей динамічних систем, заснованих на спостережуваних вхідних/вихідних даних. Особливістю цього інструменту є наявність гнучкого користувачького інтерфейсу, що дозволяє організувати дані й моделі. Бібліотека System Identification Toolbox підтримує як параметричні, так і непараметричні методи. Інтерфейс системи полегшує попередню обробку даних, роботу з ітеративним процесом створення моделей для одержання оцінок і виділення найбільш значимих даних. Що стосується математичних обчислень, то MatLab надає доступ до величезної кількості підпрограм, що містяться в бібліотеці NAG Foundation Library компанії Numerical Algorithms Group Ltd (інструмент має сотні функцій з різних областей математики, і багато з цих програм були розроблені широковідомими у світі фахівцями). Це унікальна колекція реалізацій сучасних чисельних методів комп'ютерної математики, створених за останні три десятиліття років. Лише додано до системи велику кількість документації цілком можна розглядати як фундаментальний багатотомний електронний довідник з математичного забезпечення. Сьогодні система MatLab широко використовується в техніці, науці та освіті, але все-таки вона більше підходить для аналізу даних і організації обчислень, ніж для чисто математичних викладок.

Для візуального моделювання та моделювання спільно з реальною апаратурою більш зручним є програмний пакет VisSim.

Аналітичні перетворення дозволяє виконувати більшість математичних програмних продуктів MathCad, Mathematica, Maple. З цих трьох поширених математичних пакетів найбільш потужним є Maple. **Maple** надає зручне середовище для комп'ютерних експериментів. Пакет дозволяє створювати інтегровані середовища за участю інших систем і універсальних мов програмування високого рівня. Коли розрахунки зроблені й потрібно оформити результати, то можна використовувати засоби цього пакета для візуалізації даних і підготовки ілюстрацій для публікації. Робота проходить інтерактивно – користувач вводить команди й відразу бачить на екрані результат їхнього виконання. При цьому пакет Maple зовсім не схожий на традиційне середовище програмування, де потрібна тверда формалізація всіх змінних і дій з ними. Тут же автоматично забезпечується вибір типів змінних і перевіряється коректність виконання операцій, так що в загальному випадку не потрібно опису змінних і строгої формалізації запису. Ядро символічних обчислень Maple входить до складу цілого ряду систем комп'ютерної математики – від систем для широкого кола користувачів типу MathCad – до однієї із кращих систем для чисельних розрахунків і моделювання MatLab.

На відміну від потужного та орієнтованого на високоефективні обчислення при аналізі даних пакета MatLab, програма **MathCad** – це, скоріше, редактор математичних текстів із широкими можливостями символічних обчислень і прекрасним інтерфейсом. MathCad не має мови програмування як такої, а можливості символічних обчислень запозичені з пакета Maple. Проте інтерфейс програми MathCad дуже простий, а можливості візуалізації широкі. Всі обчислення тут здійснюються на рівні візуального запису виразів у загальноживаній математичній формі. Пакет має гарні підказки, докладну документацію, цілий ряд додаткових модулів та вбудованих функцій. Однак стосовно математичних можливостей, то MathCad поступається системам Maple, Mathematica, MatLab. Не зважаючи на це, з програми MathCad випущено багато книг і навчальних курсів. Сьогодні ця система стала буквально міжнародним стандартом для технічних обчислень. Розробники Mathcad зробили все можливе, щоб користувач, який не має спеціальних знань з програмування, міг реалізувати велику кількість обчислювальних методів та досягти значного результату в області математичних розрахунків.

Звичайно, кожен з математичних пакетів має свої переваги і недоліки та є зручнішим для розв'язання конкретних завдань. В другій частині даної книги більш детально зосереджено увагу на системах MatLab та MathCad, які орієнтовані на різні типи задач і дають можливість розв'язання різних за складністю завдань. Приклади розрахунків у пакетах MatLab та MathCad наведено у частині 2 (Додатки Б і В).

1.7.2 Програмні пакети для імітаційного моделювання

Імітаційне моделювання – це метод дослідження, при якому досліджувана система замінюється моделлю, що з достатньою точністю описує реальну систему, з нею проводяться експерименти з метою одержання інформації про цю систему.

До імітаційного моделювання прибігають, коли:

- дорого або неможливо експериментувати на реальному об'єкті;
- неможливо побудувати аналітичну модель, тому що в системі є час, причинні зв'язки, наслідки, нелінійності, стохастичні (випадкові) змінні;
- необхідно зімітувати поведінку системи в часі.

Імітаційне моделювання може застосовуватися у різних сферах діяльності. Особливо ефективно моделювання при вирішенні таких завдань:

- проектування та аналіз виробничих систем;
- оцінювання різних систем озброєнь;
- визначення вимог до устаткування та протоколів мереж зв'язку;
- модернізація різних процесів у діловій сфері;
- аналіз фінансових і економічних систем.

Під **імітаційною системою** розуміють програмний або апаратно-програмний комплекс, призначений для вирішення завдань із використанням методу імітаційного моделювання. При виділенні різновидів імітаційних систем виходять із того, що вони є інструментальними засобами, які забезпечують автоматизовану підтримку певних видів діяльності користувача.

Імітаційна система реалізує алгоритм вирішення завдання і надає користувачеві сервісні можливості з керування обчислювальним процесом. Автоматизована підтримка інших етапів системного аналізу засобами імітаційної системи не є обов'язковою. Однак саме ступінь їхньої автоматизації визначає можливості імітаційної системи і є основою їхньої класифікації. З урахуванням етапності системного аналізу і технологічних завдань, що розв'язуються на них, виділимо можливий набір засобів імітаційної системи, що автоматизують виконання ряду функцій, реалізованих на цих етапах.

1. *Створення моделі* може бути підтримано такими засобами автоматизації:

- а) частково готовою моделлю або моделями;
- б) компіляторами з алгоритмічної мови високого рівня, спеціалізація яких полегшує процес складання алгоритмів імітації;
- в) спеціальною мовою високого рівня, що дозволяє виконати інформаційний або математичний опис моделі системи;

г) конверторами моделей, що дозволяють здійснювати перетворення моделей одного виду в моделі іншого виду (інформаційної в математичну, математичної в імітаційну, інформаційної в імітаційну);

д) засобами контролю погодженості різних видів моделей з концептуальним поданням моделі.

2. *Перевірка адекватності та технічної реалізованості* може виконуватися з використанням:

а) програм обчислення показників адекватності;

б) автоматизованої технології проведення обмеженого експерименту з імітаційною моделлю;

в) програм обчислення характеристик складності моделі;

г) програм обчислення ресурсних показників методу рішення завдання.

3. *Корекція моделі* може забезпечуватися:

а) автоматизованими технологіями редагування текстів моделей;

б) програмами еквівалентних перетворень математичних і алгоритмічних моделей заданого класу.

4. *Створення алгоритму вирішення завдання* може підтримуватися:

а) методоорієнтованими бібліотеками та пакетами програм;

б) конструкторами алгоритмів рішення завдань;

в) інформаційними системами підтримки прийняття рішень тощо.

5. *Складання та уточнення схеми вирішення завдання* може виконуватися з використанням:

а) програмних засобів контролю інформаційної сумісності сімейства завдань;

б) редакторів текстів обчислювальних схем тощо.

При виборі засобів імітаційного моделювання варто враховувати всі можливості, що їх вони надають, які можна об'єднати в групи:

– основні характеристики;

– спільне програмне забезпечення;

– анімація;

– статистичні можливості;

– звіти з вихідними даними і графіками;

– послуги, що надаються замовникам, і документація.

Найбільш популярними пакетами імітаційного моделювання є:

1. Arena компанії Rockwell Automation;

2. AnyLogic компанії XJ Technologies;

3. GPSS World фірми Minuteman Software;

4. Process Charter 1.0.2 компанії Scitor;

5. Powersim 2.01 фірми Modell Data AS;

6. Ithink 3.0.61 виробництва High Performance Systems;

7. Extend+BPR 3.1 компанії Imagine That!;

8. Vensim фірми Ventana Systems.

Ці пакети найбільше відрізняються стилем моделювання, тобто середовищем, за допомогою якого створюються моделі. У пакеті Process Charter модель будується за допомогою блок-схеми. Powersim і Ithink використовують систему позначень Systems Dynamics, запропоновану в 1961 р. Джеєм Форрестером Массачусетського технологічного інституту. Extend застосовує компоновальні блоки. Всі продукти, крім Process Charter, дозволяють проводити аналіз чутливості, тобто багаторазово виконувати модель із різними вхідними параметрами, щоб порівняти результати декількох прогонів.

Більш детально розглянемо найбільш поширені пакети для імітаційного моделювання.

GPSS World (GPSSW, General Purpose System Simulation World – Світова загальноцільова система моделювання), розроблена для ОС Windows. Цей програмний продукт увібрав в себе весь арсенал новітніх інформаційних технологій. Він містить розвинені графічні оболонки для створення моделей та інтерпретації вихідних результатів моделювання, засоби мультимедіа та відео, об'єктно-орієнтоване програмування та ін. В основу системи GPSS World покладена мова імітаційного моделювання GPSS (General Purpose System Simulation – Загальноцільова система моделювання).

Система GPSS World – потужне універсальне середовище моделювання як дискретних, так і неперервних процесів, призначене для професійного моделювання найрізноманітніших процесів і систем. За допомогою цієї системи, наприклад, можна ефективно моделювати як виробничі, так і невиробничі процеси: функціонування торговельних і розважальних закладів, портів, вуличний рух, проведення воєнних дій, роботу редакцій, установ і мережі Internet, різних систем масового обслуговування тощо. Система має великий набір команд для керування процесом моделювання, які можна як використовувати в інтерактивному режимі, так і вносити в модель. Забезпечено можливість проведення експериментів, згенерованих системою, користувацьких і оптимізаційних. У системі GPSSW реалізована процедура візуалізації процесу функціонування моделі з використанням методів мультиплікації. Також система GPSSW має новий високошвидкісний транслятор, що працює в сотні раз швидше його попередників. Для швидкого виправлення помилок використовується повноекранний текстовий редактор.

Система GPSSW досить проста у вивченні і універсальна в застосуванні. Ефективне використання системи передбачає виконання ряду етапів.

1. Постановка задачі.
2. Виявлення основних особливостей.
3. Створення імітаційної моделі процесу.
4. Подання імітаційної моделі в системі GPSSW.

5. Моделювання системи.

Як приклад розглянемо модель одноканальної розімкнутої системи масового обслуговування (СМО). На рис. 1.10 показана модель найпростішої системи масового обслуговування, у якій виділені основні події.

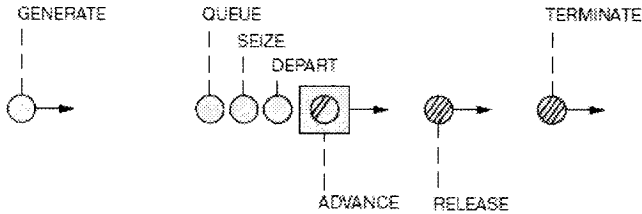


Рисунок 1.10 – Найпростіша система масового обслуговування

Охарактеризуємо кожну подію, що виникла в СМО.

1. Поява вимоги в системі (GENERATE – Генерувати).

2. Вхід вимоги в чергу (QUEUE – Черга).

3. Визначення зайнятості каналу обслуговування (SEIZE – Зайняти).

Якщо канал зайнятий, то вимога залишається в черзі, якщо вільний – то входить у канал обслуговування.

4. Вихід вимоги із черги (DEPART – Вийти).

5. Обслуговування вимоги в каналі обслуговування (ADVANCE – Затримати).

6. Звільнення каналу обслуговування (RELEASE – Звільнити).

7. Вихід вимоги із системи (TERMINATE – Завершити).

Ця програма в системі GPSSW буде виглядати так, як показано на рис. 1.11. Використані в програмі оператори повністю відповідають змісту і логіці системи, що моделюється. Праворуч від операторів пишуться параметри (ознаки, змінні), які характеризують дану подію. Так, в операторі GENERATE перша цифра 7 – визначає середній інтервал часу між надходженнями вимог у систему на обслуговування, а друга 2 – максимально припустиме відхилення від середнього часу. В операторах QUEUE і DEPART цифра 1 визначає номер черги, в яку ввійшла і з якої має вийти вимога. В операторах SEIZE і RELEASE символи KAN визначають символічне ім'я каналу обслуговування, у який збирається ввійти вимога, якщо він звільнився, і вийти – якщо вимога вже в ньому закінчила обслуговування. В операторі ADVANCE перша цифра 6 – визначає середній час обслуговування вимоги, а друга 3 – максимально припустиме відхилення від цього часу. Оператор TERMINATE виконує видалення однієї вимоги із системи. Цифра 200 в операторі START означає число вимог, які необхідно пропустити через систему.

В цьому прикладі вимоги надходять на обслуговування в систему випадково в інтервалі [5-9] одиниць часу з рівномірним розподілом. А час обслуговування коливається в інтервалі [3-9] одиниць часу, також з рівномірним розподілом. При цьому моделюється проходження через систему 200 вимог.

| Untitled Model 1 | | |
|------------------|-----|--|
| GENERATE | 7,2 | : Генеруєт вхідні вимоги в систему з інтервалом [5 - 9] од. времени. |
| QUEUE | 1 | : Увеличивает поддерживаемое очередь под номером 1 на одно требование. |
| SEIZE | KAN | : Открывает занятость канала по имени KAN. |
| DEPART | 1 | : Уменьшает поддерживаемое очереди под номером 1 на одно требование. |
| ADVANCE | 6,3 | : Обслуживает требование с каналом обслуживания [3 - 9] од. времени. |
| RELEASE | KAN | : Освобождает канал обслуживания по имени KAN. |
| TERMINATE | 1 | : Выход требования из системы. |
| START | 200 | : Начало моделирования с числом требований = 200. |

Рисунок 1.11 – Вікно з введеною в нього моделлю одноканальної розімкнутої СМО

Пакет імітаційного моделювання Rockwell Arena

Система Arena компанії Rockwell Automation є поширеною на ринку програм імітаційного моделювання. В Arena використовується процесор і мова імітаційного моделювання SIMAN. Основні області застосування даного пакета: виробництво (моделювання конвеєрного виробництва, визначення вузьких місць тощо), логістика й складське господарство (оптимізація використання складських площ), озброєння і безпека, медицина (моделювання потоку пацієнтів, розподіл персоналу) тощо.

Arena надає користувачеві зручний графічний інтерфейс із набором шаблонів моделювальних конструкцій. Для створення моделі в пакеті Arena моделювальні конструкції спочатку перетягують у вікно моделі, а потім з'єднують, щоб позначити рух об'єктів у системі. Потім моделюючі конструкції деталізуються за допомогою діалогових вікон або вбудованих таблиць. В ієрархії моделі може бути необмежене число рівнів. Rockwell Arena випускається тільки для операційної системи Windows. В Arena передбачений експорт даних з Microsoft Excel і Microsoft Access.

Число потоків випадкових чисел у пакеті Arena не обмежено. Більше того, користувач має доступ до 12 стандартних теоретичних розподілів ймовірностей, а також до емпіричних розподілів.

Arena забезпечує виведення на екран двовимірної й тривимірної (Arena 3DPlayer) анімації й дозволяє виводити на екран динамічну графіку (гістограми та графіки тимчасової залежності).

Пакет імітаційного моделювання AnyLogic

AnyLogic – програмне забезпечення для імітаційного моделювання складних систем і процесів, розроблене російською компанією XI Technologies.

Графічне середовище AnyLogic побудовано за тим ж принципом, що і у Rockwell Arena. Моделювальні конструкції розташовуються в палітрах (аналог шаблонів в Arena). Для створення моделі, як і в Arena, моделювальні конструкції перетягують в область моделі і з'єднують. Деталізувати моделювальні конструкції можна, виділивши їх і змінивши параметри, використовуючи панель властивостей. AnyLogic підтримує ієрархічне моделювання, а також створення власних моделювальних конструкцій і об'єднання їх у бібліотеки (тільки для версії Professional). AnyLogic заснований на Java і базується на платформі Eclipse – сучасному стандарті для бізнесів-додатків. Завдяки Eclipse AnyLogic працює на всіх поширених операційних системах (Windows, Mac, Linux і т. д.).

У редакторі AnyLogic можна розробити анімацію та інтерактивний графічний інтерфейс моделі. Анімація може бути ієрархічною і підтримувати кілька перспектив. Наприклад, можна визначити глобальний погляд на процес виробництва, а також детальні анімації конкретних операцій, а потім перемикаєтися між ними.

В AnyLogic користувачеві доступно 29 стандартних теоретичних розподілів. Є можливість зафіксувати набір випадкових чисел і зробити абсолютно ідентичний експеримент.

Для створення звітів в AnyLogic відведена спеціальна палітра «Статистика», у якій утримуються конструкції для збору даних в ході роботи моделі. У цій палітрі також знаходяться різні діаграми, графіки та гістограми.

Для порівняння систем на практиці були побудовані дві ідентичні моделі (для AnyLogic і для Arena) мережі передачі даних. Процеси побудови моделі в AnyLogic і в Arena мало відрізняються, як і зовнішній вигляд моделей.

У поданій моделі за основу прийняті пакети даних, джерелами яких є комп'ютери віддалених користувачів. У моделі є ресурси мережесх адаптерів і маршрутизатора. Ємність маршрутизатора дорівнює 1, тобто в кожний момент часу маршрутизатор може обробляти тільки один пакет.

За допомогою даної моделі можна оцінити максимальне навантаження мережі із заданою пропускнуною спроможністю.

Прогін моделі на AnyLogic виявився в декілька разів швидшим, ніж в Arena. Після прогону моделі Arena вивела на екран результати моделювання у вигляді звіту, що містить таку інформацію: середній, максимальне й мінімальний час перебування пакетів даних; середні довжини черг; завантаженість ресурсів тощо. Прогін моделі в AnyLogic здійснюється в окремому вікні у вигляді презентації. Єдиною вбудованою статистикою, що збирається для черг в AnyLogic, є середня довжина черги; для ресурсів – завантаження. Для одержання додаткових даних про систему необхідно створювати власні збирачі статистики.

З рис. 1.12 видно, що в Arena, на відміну від AnyLogic (рис. 1.13), немає окремої конструкції для черг. Черги створюються разом з ресурсами обмеженої ємності.

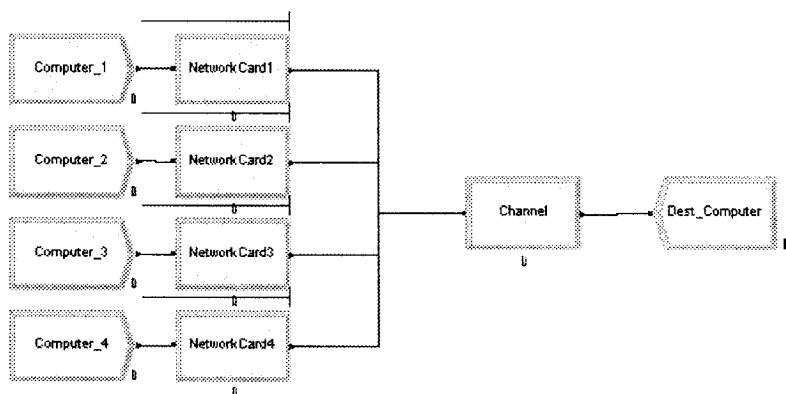


Рисунок 1.12 – Модель в Arena

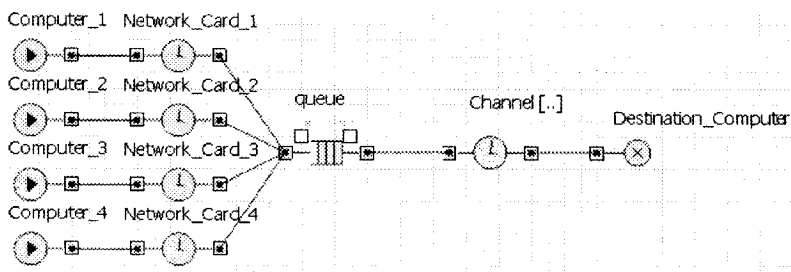


Рисунок 1.13 – Модель в AnyLogic

Крім розглянутих спеціалізованих пакетів для імітаційного моделювання пакет розширення MatLab Simulink for Windows також служить для імітаційного моделювання моделей, що складаються з графічних блоків із заданими властивостями (параметрами). Компоненти моделей, у свою чергу, є графічними блоками і моделями, які містяться в ряді бібліотек і за допомогою миші можуть переноситися в основне вікно та з'єднуватися один з одним необхідними зв'язками. До складу моделей можуть входити джерела сигналів різного виду, віртуальні реєструвальні прилади, графічні засоби анімації. Подвійне клацання мишею на блоці моделі виводить вікно зі списком його параметрів, які користувач може змінювати. Запуск імітації забезпечує математичне моделювання побудованої моделі з наочним візуальним поданням результатів.

Пакет заснований на побудові блокових схем шляхом перенесення блоків з бібліотеки компонентів у вікно редагування створюваної користувачем моделі. Потім модель запускається на виконання. На рис. 1.14 показаний процес моделювання простої системи – гідравлічного циліндра. Контроль здійснюється за допомогою віртуальних осцилографів – на рис. 1.15 видно екрани двох таких осцилографів і вікно простої підсистеми моделі. Можливе моделювання складних систем, що складаються з безлічі підсистем. Simulink складає і вирішує рівняння стану моделі і дозволяє підключати в потрібні точки різноманітні віртуальні вимірювальні прилади.

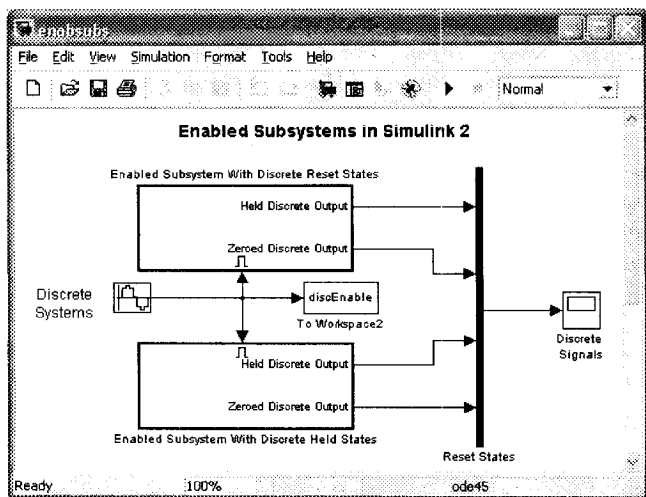


Рисунок 1.14 – Процес моделювання гідравлічного циліндра

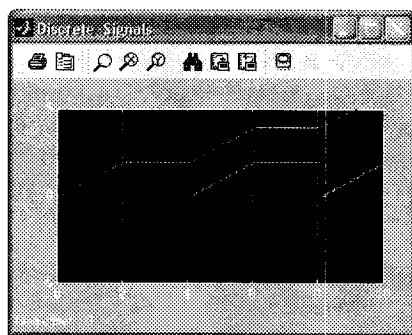


Рисунок 1.15 – Приклад моделювання системи гідравлічного циліндра за допомогою розширення Simulink

1.7.3 Системи автоматизованого проектування

У сучасному виробництві широке поширення одержали **системи автоматизованого проектування (САПР, computer aided design)**, які дозволяють проектувати технологічні процеси з меншими витратами часу та засобів, зі збільшенням точності спроектованих процесів і програм обробки, що скорочує витрати матеріалів та час обробки завдяки тому, що режими обробки також розраховуються та оптимізуються за допомогою ЕОМ.

Технічне забезпечення САПР засновано на використанні обчислювальних мереж і телекомунікаційних технологій, персональних комп'ютерів та робочих станцій.

Математичне забезпечення САПР характеризується різноманітністю методів обчислювальної математики, статистики, математичного програмування, дискретної математики, штучного інтелекту. Програмні комплекси САПР відносять до числа найбільш складних сучасних програмних систем, заснованих на операційних системах Unix, Windows, мовах програмування C, C++, Java і інших, сучасних CASE-технологіях, реляційних і об'єктно-орієнтованих системах керування базами даних (СКБД), стандартах відкритих систем і обміну даними в комп'ютерних середовищах.

Проектування, при якому всі проектні рішення або їхню частину одержують шляхом взаємодії людини та ЕОМ, називають *автоматизованим* на відміну від ручного (без використання ЕОМ) або *автоматичного* (без участі людини на проміжних етапах). Система, що реалізує автоматизоване проектування, являє собою систему автоматизованого проектування (в англійській мові написанні CAD System – Computer Aided Design System). САПР (або CAD) звичайно використовується разом із системами автоматизації інженерних розрахунків і аналізу САЕ (Computer-Aided engineering). Дані із CAD-систем передаються в САМ (Computer-Aided manufacturing) – систему автоматизованої розробки програм обробки деталей для верстатів.

САЕ – автоматизоване конструювання, використання спеціального програмного забезпечення для проведення інженерного аналізу міцності та інших технічних характеристик компонентів, виконаних у системах автоматизованого проектування. Програми автоматизованого конструювання дозволяють здійснювати динамічне моделювання, перевірку та оптимізацію виробів і засобів їхнього виробництва.

САМ – автоматизоване виробництво. Термін використовується для позначення програмного забезпечення, основною метою якого є створення програм для керування верстатами з ЧПК (числове програмне керування). Вхідними даними САМ-системи є геометрична модель виробу, розроблена в системі автоматизованого проектування. У процесі інтерактивної роботи

із тривимірною моделлю в САМ-системі інженер визначає траєкторії руху різального інструменту по заготовці виробу, які потім автоматично верифікуються, візуалізуються (для візуальної перевірки коректності) і обробляються постпроцесором для одержання програми керування конкретним верстатом.

Структура САПР. САПР складається з проектувальної і обслуговувальної підсистем. Проектувальні підсистеми безпосередньо виконують проектні процедури. Прикладами проектувальних підсистем можуть слугувати підсистеми геометричного тривимірного моделювання механічних об'єктів, виготовлення конструкторської документації, схемотехнічного аналізу, трасування з'єднань у друкованих платах.

Обслуговувальні підсистеми забезпечують функціонування проектувальних підсистем, їхню сукупність часто називають системним середовищем (або оболонкою) САПР. Типовими обслуговувальними підсистемами є підсистеми керування проектними даними (PDM – Product Data Management), керування процесом проектування (DesPM – Design Process Management), користувацького інтерфейсу для зв'язку розробників з EOM, CASE (Computer Aided Software Engineering) для розробки та супроводу програмного забезпечення САПР, навчальні підсистеми для освоєння користувачами технологій, реалізованих у САПР. На сьогодні створено велику кількість програмно-методичних комплексів для САПР із різними ступенем спеціалізації й прикладною орієнтацією. У результаті автоматизація проектування стала необхідною складовою частиною підготовки інженерів різних спеціальностей; інженер, що не має знань та не вміє працювати в САПР, не може вважатися повноцінним фахівцем.

Огляд найбільш поширених САПР світових виробників.

AutoCAD – найвідоміший із продуктів компанії Autodesk, універсальна система автоматизованого проектування, що поєднує у собі функції двовимірного креслення й тривимірного моделювання. З'явилася в 1982 році і була однією з перших САПР, розроблених для PC. Швидко завоювала популярність серед проектувальників, інженерів і конструкторів різних галузей промисловості завдяки демократичним цінам.

AutoCAD прискорює щоденну роботу зі створення креслень, підвищує швидкість і точність їхнього виконання. Середовище концептуального проектування забезпечує легке й інтуїтивне створення та редагування твердих тіл і поверхонь. AutoCAD дозволяє легко й швидко створювати на основі моделі розрізи й проекції, ефективно формувати комплекти креслень і керувати ними: групувати їх за розділами проекту та іншими логічними категоріями, створювати переліки аркушів, керувати видами креслень, архівувати комплекти проектної документації та організовувати спільну роботу фахівців. Наявні в AutoCAD засоби візуалізації, такі як анімація й реалістичне тонування, допомагають виявити будь-які вади на

ранніх етапах проектування.

Формат DWG, що використовується в AutoCAD, є стандартом серед проектувальників різних галузей промисловості, крім того, є можливості експорту й імпорту інших розповсюджених файлових форматів, таких як pdf, що дозволяє ефективно організувати обмін даними між фахівцями.

Програма постійно розвивається, серед можливостей, що з'явилися нещодавно, можна назвати параметричні взаємозв'язки між об'єктами, створення та редагування об'єктів довільної форми тощо. Існують спеціалізовані галузеві різновиди AutoCAD для архітектури, дорожнього будівництва та землевпорядження, електротехніки, машинобудування тощо. Для фахівців, яким не потрібні функції роботи з 3D-графікою, існує полегшена версія AutoCAD, призначена для створення двовимірних креслень – AutoCAD LT.

Рішення Autodesk для промислового виробництва й машинобудування засновані на технології цифрових прототипів, тобто надають конструкторам, інженерам, дизайнерам і технологам можливість повністю досліджувати виріб ще на етапі проектування. За допомогою даної технології виробники створюють цифрові моделі та проекти, конструюють, перевіряють, оптимізують і керують ними на всіх етапах – від ідеї до реального втілення.

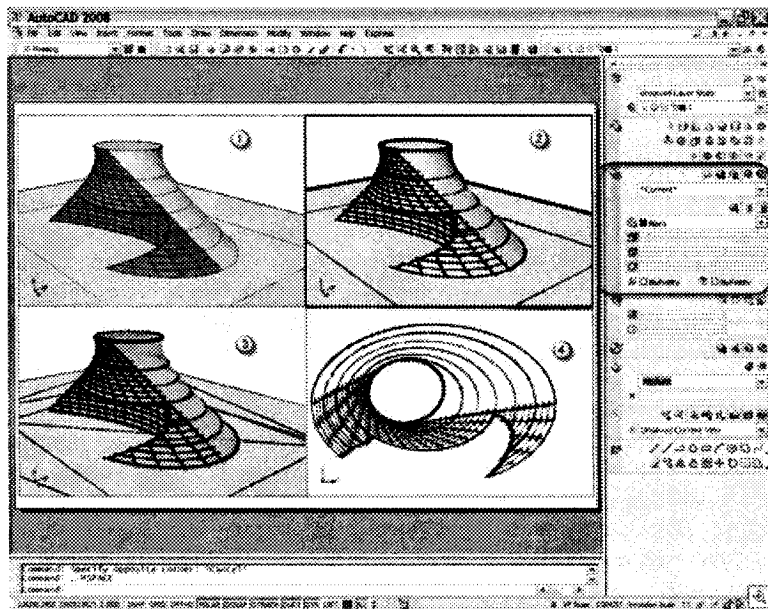


Рисунок 1.16 – Інтерфейс AutoCAD з прикладом 3D-моделювання

AutoCAD Mechanical – продукт на платформі AutoCAD для промислового виробництва, що є частиною технології цифрових прототипів Autodesk. Він допомагає прискорити процес проектування, дозволяючи в той же час використовувати досвід і проекти, накопичені при роботі в AutoCAD. Маючи у своєму складі бібліотеки ДСТУ, стандартних деталей і функції автоматизації типових завдань, він забезпечує значний вигравш у продуктивності при проектуванні.

AutoCAD Electrical – це AutoCAD для проектування електричних систем керування, що є важливою частиною технології цифрових прототипів Autodesk і що дозволяє працювати швидко, якісно й зі значно меншими витратами в знайомому середовищі проектування. Спеціалізовані функції й великі бібліотеки умовних позначень дозволяють підвищити продуктивність, усунути ризик виникнення помилок і забезпечити точність інформації, переданої у виробництво.

AutoCAD Inventor Suite являє собою збалансований набір рішень Autodesk для проектування та конструювання в промисловому виробництві. Рішення поєднують у собі інтуїтивне середовище 3D моделювання деталей і виробів з інструментами, дозволяють конструкторам зосередитися на функціональних вимогах до проекту.

CATIA – система автоматизованого проектування французької фірми Dassault Systems. CATIA V1 була анонсована в 1981 році. У даний момент у світі використовуються дві версії – V4 і V5, які значно відрізняються між собою. CATIA V4 була анонсована в 1993 році й створювалася для Unix-подібних операційних систем, CATIA V5 була анонсована в 1998 році, і це перша з версій, що може працювати під керуванням Microsoft Windows. За запевненнями Dassault Systems, CATIA V5 була написана «з нуля» і втілила в собі передові технології САПР. Спочатку CATIA V5 не користувалася особливою популярністю на ринку, і щоб стимулювати її використання Dassault Systems висунула концепцію PLM (Product Lifecycle Management). Ідея PLM виявилася вдалою і її підхопила майже вся індустрія САПР. У лютому 2008 року Dassault Systems анонсувала нову версію системи – CATIA V6. V6 буде підтримувати програми моделювання для всіх інженерних дисциплін і колективні бізнес-процеси протягом життєвого циклу виробу. Нова концепція фірми одержала назву «PLM 2.0 на платформі V6». Суть концепції – тривимірне моделювання і колективна робота в реальному часі. Для зв'язку між людьми, що перебувають у різних точках світу, передбачені засоби простого підключення до Web. PLM 2.0 – це новий підхід, що відкриває можливість використовувати інтелектуальні результати онлайн-взаємозв'язку. Кожний користувач може придумувати, розробляти продукти та обмінюватися інформацією на універсальній 3D-мові. Користувачі зможуть у наочній формі оперувати одночасно віртуальними та реальними об'єктами.

Pro/Engineer – CAD система високого рівня. Містить у собі всі необхідні модулі для твердотілого моделювання деталей і створення креслярської документації. Має вбудовані можливості для проектування зварних конструкцій.

SolidWorks – продукт компанії SolidWorks Corporation, система автоматизованого проектування у трьох вимірах, працює під керуванням Microsoft Windows. Розроблена як альтернатива для двовимірних програм САПР. Набула популярності завдяки простому інтерфейсу. Основний продукт SolidWorks містить інструменти для тривимірного моделювання, створення креслень, роботи з листовим металом, зварними конструкціями і поверхнями довільної форми. Є можливість імпортування великої кількості файлів 2D і 3D CAD-програм. Є API для програмування в середовищі Visual Basic і C. Також є програма для аналізу методом кінцевих елементів початкового рівня CosmosXpress.

ADEM (Automated Design Engineering Manufacturing) – російська інтегрована CAD/CAM/CAPP-система, призначена для автоматизації конструкторсько-технологічної підготовки виробництва (КТПВ). ADEM була створена як єдиний продукт, що містить в собі інструментарій для проєктантів і конструкторів (CAD), технологів (CAPP – Computer-Aided Process Planning) і програмістів ЧПК. Тому вона містить декілька різних предметно-орієнтованих САПР під єдиною логікою керування і на єдиній інформаційній базі. ADEM дозволяє автоматизувати такі види робіт: 3D і 2D моделювання та проектування; оформлення проектно-конструкторської й технологічної документації; проектування технологічних процесів; аналіз технологічності й нормування проєкту; програмування устаткування. ADEM застосовується у різних галузях: авіаційній, атомній, аерокосмічній, машинобудівній, металургійній, верстатобудівній та інших.

T-FLEX CAD – система автоматизованого проектування, розроблена компанією «Топ Системи» з можливостями параметричного моделювання і наявністю засобів оформлення конструкторської документації відповідно до стандартів серії ЕСКД (Єдина система конструкторської документації). T-FLEX CAD є ядром комплексу T-FLEX CAD/CAM/CAE/CAPP/PDM – набору засобів для вирішення завдань технічної підготовки виробництва в різних галузях промисловості. Комплекс поєднує системи для конструкторського і технологічного проектування, модулі підготовки програм керування для верстатів та інженерних розрахунків. Всі програми комплексу функціонують на єдиній інформаційній платформі системи технічного документообігу і ведення складу виробів.

вCAD – дво- і тривимірна система автоматизованого проектування, розроблена російською компанією «ПРОПРО Група». вCAD являє собою інтегрований пакет для двовимірного креслення, об'ємного моделювання й реалістичної візуалізації (рис. 1.17).

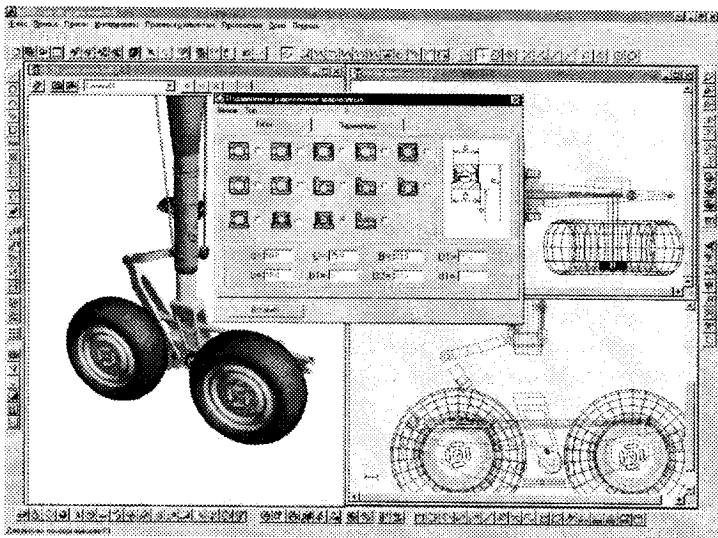


Рисунок 1.17 – Інтерфейс bCAD з прикладом 3D-моделювання

Система одержала широке поширення в меблевому виробництві та дизайні інтер'єрів. Незважаючи на досить розвинені засоби проектування, у промисловості практично не застосовується.

КОМПАС – система автоматизованого проектування, розроблена російською компанією «АСКОН» з можливостями оформлення проектної й конструкторської документації відповідно до стандартів серії ЕСКД і СПДБ (Система проектної документації для будівництва). Існує у двох версіях: Компас-Графік і КОМПАС-3D, відповідно призначених для плоского креслення і тривимірного проектування.

MechaniCS – додаток до AutoCAD або Autodesk Inventor, призначений для оформлення креслень відповідно до ЕСКД, проектування систем гідропневмоелементів, зубчастих зачеплень, валів, інженерного аналізу, розрахунку розмірних ланцюгів, створення користувацьких бібліотек. MechaniCS забезпечує фахівця всім необхідним для проектування машинобудівних об'єктів: більш ніж 1500 стандартами (включаючи ДСТУ, ОСТ, ДІ і ISO) і уніфікованими компонентами, можливістю створювати власні інтелектуальні об'єкти, виконувати інженерні розрахунки з відображенням результатів на моделі, оформляти проекції креслень за ЕСКД і багато іншого. MechaniCS дає конструкторові можливість застосовувати не тільки геометричні параметри стандартних елементів, але й їхні механічні властивості. На об'єкти в складальних кресленнях (при використанні AutoCAD) можна накладати геометричні і параметричні залежності, використовувати попередньо встановлені залежності при їхньому розміщенні на кресленні.

Контрольні запитання та завдання

1. Що таке модель? В чому суть процесу моделювання?
2. При моделюванні об'єкт відображується у моделі. Чи існує обернене відображення?
3. Як визначається адекватність моделі, чим вона характеризується?
4. Перерахуйте причини можливої неадекватності моделей?
5. Як ви розумієте вираз «модель адекватна реальному процесу»?
6. Дайте означення основних характеристик точності моделі?
7. Що таке обчислювальний експеримент? Як він реалізується?
8. Наведіть приклади типових задач обчислювальної математики. Класифікуйте їх за видом математичного апарату та за фізичною сутністю. Покажіть застосування цих методів для розв'язання прикладних задач автоматизації та управління.
9. Що є джерелами похибок моделювання?
10. Чим відрізняються локальна та глобальна похибки?
11. Якими якісними показниками характеризується модель?
12. З якої послідовності етапів складається процес побудови математичної моделі?
13. Якими можуть бути види вхідних параметрів математичної моделі? Приведіть приклади.
14. Дайте означення гомеоморфізму та ізоморфізму.
15. Для вирішення яких задач може бути використана математична модель?
16. Що розуміють під поняттям коректна модель? Які умови повинна задовольняти коректна модель?
17. Який вид моделей є ізоморфним оригіналу?
18. В чому полягає принципова різниця між аналітичними та імітаційними моделями?
19. Інструменти імітаційного моделювання, призначення та класифікація. Сучасні тенденції в імітаційному моделюванні.
20. Які моделі використовуються для прогнозування процесів на фінансовому та фондовому ринках?
21. Проведіть аналіз та класифікацію декількох математичних моделей в цікавій для вас області, опишіть особливості кожного етапу моделювання. Встановіть аналоги розглянутих математичних моделей в інших областях.
22. Сформулюйте декілька варіантів змістовних постановок задач моделювання роботи: продовольчого магазину; регульованого перехрестя доріг; факультету вищого навчального закладу тощо.
23. Розгляньте процедуру комп'ютерного моделювання починаючи з математичного опису до алгоритму та програми для заданої задачі: кулька з'являється на екрані комп'ютера всередині прямокутника в

- точці (x_c, y_c) і рухається до досягнення однієї зі «стінок» прямокутника зі швидкістю \vec{v} (v_x, v_y) (v_x і v_y – горизонтальна та вертикальна складові вектора швидкості, x_c, y_c – координати центра кульки в даний момент часу).
24. Доповнимо завдання 23 ще однією умовою: якщо кулька досягає краю прямокутника, то вона відбивається від стінки й продовжує рух.
 25. Розгляньте процедуру комп'ютерного моделювання, починаючи з математичного опису до алгоритму та програми для заданої задачі: дві однакові кульки рухаються на екрані, вдаряючись одна в одну вони змінюють напрямок руху.
 26. Наведіть приклади сучасних САПР.
 27. Проведіть порівняння можливостей поширених програмних систем для математичних обчислень.
 28. Охарактеризуйте етапи реалізації імітаційних моделей на комп'ютерах на прикладі відомих систем імітаційного моделювання.

Необхідною і достатньою умовою існування єдиного розв'язку є нерівність нулю визначника A (лінійна незалежність рівнянь):

$$\det A \neq 0.$$

Методи розв'язання систем лінійних рівнянь можна розділити на прямі та ітераційні. До прямих, які дозволяють одержати точний розв'язок, відносять методи визначників Крамера, Гаусса, прогонки. Ітераційні методи, що ґрунтуються на одержанні і уточненні послідовних наближень до точного розв'язку, ефективні в тому випадку, коли є багато нульових коефіцієнтів або високий порядок системи (метод Гаусса ефективний до порядку 10^4 , ітераційні – до 10^6).

2.1.1 Прямі методи

До найбільш популярних прямих методів відносять метод Гаусса та його різновиди, метод Крамера (визначників), метод оберненої матриці, а також метод прогонки, що використовується в задачах з діагональними матрицями. Проте метод Крамера (визначників), детально розглянутий в стандартних курсах вищої математики, не може бути застосований в більшості практичних задач через велику складність розрахунку визначників навіть при невеликому зростанні порядку системи. Тому в цьому розділі буде зосереджено увагу на розгляді методу Гаусса, який, якщо й поступається ітераційним методам в певних практичних областях, все ж таки є найбільш універсальним, а також методу прогонки, що використовується в задачах з діагональними матрицями.

2.1.1.1 Метод Гаусса

Цей метод є одним з найпоширеніших методів розв'язання СЛАР. У його основі лежить ідея послідовного виключення невідомих, що приводить вихідну систему до трикутного виду, у якому всі коефіцієнти нижче головної діагоналі дорівнюють нулю. Існують різні обчислювальні схеми, що реалізують цей метод. Найбільше поширення мають схеми з вибором головного елемента по рядку, по стовпцю, або по всій матриці.

Класичний метод Гаусса (метод виключення) ґрунтується на приведенні матриці коефіцієнтів системи (2.1) до трикутного вигляду:

$$\begin{pmatrix} * & * & * & \dots & * & * \\ 0 & * & * & \dots & * & * \\ 0 & 0 & * & \dots & * & * \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & * & * \\ 0 & 0 & 0 & \dots & 0 & * \end{pmatrix}$$

$$b_i^{k+1} = \begin{cases} b_i^k, & i \leq k \\ b_{k+1}^k / a_{k+1, k+1}^k, & i = k+1 \\ b_i^k - b_{k+1}^k \cdot a_{i, k+1}^k, & i > k+1 \end{cases} \quad (2.4)$$

При програмуванні методу для зберігання інформації про коефіцієнти матриць достатньо використовувати один двовимірний масив з n рядками та $n+1$ стовпцем, в якому розміщується розширена матриця системи $(A|B)$.

Умови завершення прямого ходу методу Гаусса

При виконанні розрахунків може виникнути проблема: елемент $a_{k+1, k+1}^k = 0$. Ця ситуація виникає, коли всі елементи матриці A^k у рядках $i > k$ дорівнюють нулю. Система в цьому випадку має вигляд:

$$\begin{cases} x_1 + a_{12}^k x_2 + \dots + a_{1n}^k x_n = b_1^k \\ 0 + a_{22}^k x_2 + \dots + a_{2n}^k x_n = b_2^k \\ \dots \dots \dots \\ 0 + \dots 0x_{k-1} + a_{kk}^k x_k + \dots + a_{kn}^k x_n = b_k^k \\ 0 + \dots 0x_{k-1} + 0x_k + 0x_{k+1} + \dots + 0x_n = b_{k+1}^k \\ \dots \dots \dots \\ 0 + \dots 0x_{k-1} + 0x_k + 0x_{k+1} + \dots + 0x_n = b_n^k \end{cases} \quad .$$

Застосування формул (2.3), (2.4) неможливе. В даному випадку система має нескінченну множину розв'язків, або не має їх зовсім. Це визначається з матриці B^k .

Якщо всі елементи $b_i^k = 0$ при $i \geq k+1$, то система має нескінченну множину розв'язків, причому корені x_1, \dots, x_k називаються залежними і виражають через значення x_{k+1}, \dots, x_n , які називають незалежними коренями.

Якщо хоч один елемент $b_i^k \neq 0$ при $i \geq k+1$, то розв'язків у системи немає.

У випадку, коли проблеми ділення на нуль не виникало і була отримана трикутна матриця A^n , система має єдиний розв'язок. Для його пошуку застосовується **зворотний хід методу Гаусса**:

$$\begin{aligned} x_n &= b_n^n \\ x_{n-1} &= b_{n-1}^n - \sum_{k=1}^i a_{n-k+1}^n \cdot x_{n-k+1}, \quad i = \overline{1, n-1} \end{aligned} \quad (2.5)$$

Алгоритм методу наведений на рисунку 2.1. Приклад розв'язання СЛАР методом Гаусса наведено у Додатку А (2-а частина посібника).

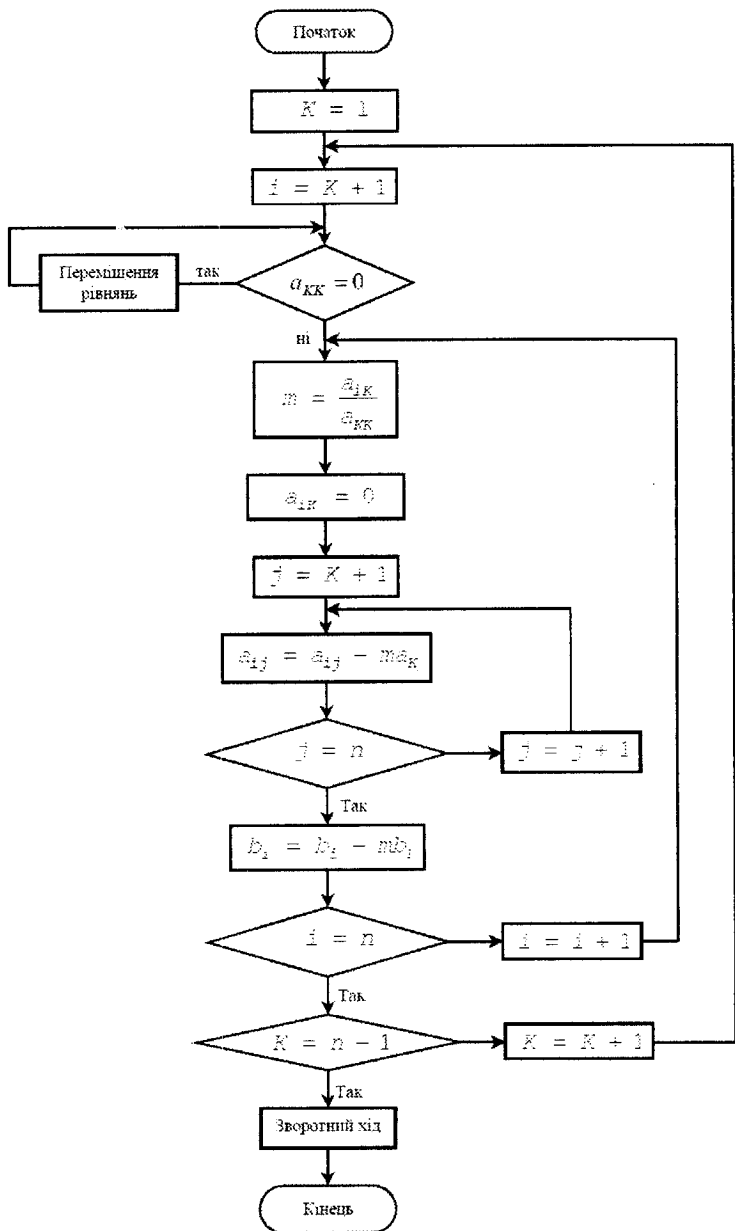


Рисунок 2.1 – Алгоритм методу Гаусса

2.1.1.2 Метод виключення Гаусса-Жордана

Цей метод дозволяє привести матрицю коефіцієнтів до діагонального вигляду. Єдиною його формальною відмінністю від попереднього методу є те, що замість $i > k$ підставляється $i \neq k$ (k -й рядок називається провідним). В методі Гаусса перетворення торкалися тільки рівнянь, що стоять нижче провідного рядка. Від класичного методу Гаусса цей метод відрізняється тим, що пронормований рядок віднімається не тільки від нижніх, а й від верхніх рівнянь системи, в результаті чого матриця A зводиться до одиничної. Зворотний хід в такому разі не потрібен, розв'язком системи буде стовпець B^n . Метод Гаусса-Жордана можна використовувати з вибором головного елемента чи без. Пошук головного елемента відбувається аналогічно попередньому методу. Формули (2.3), (2.4) з вказаними змінами будуть мати вигляд:

$$a_{ij}^{k+1} = \begin{cases} a_{ij}^k, & j \leq k \\ a_{k+1,j}^k / a_{k+1,k+1}^k, & i = k+1, j > k \\ a_{ij}^k - a_{k+1,j}^{k+1} \cdot a_{i,k+1}^k, & i \neq k+1, j > k \end{cases}, \quad (2.6)$$

$$b_i^{k+1} = \begin{cases} b_{k+1}^k / a_{k+1,k+1}^k, & i = k+1 \\ b_i^k - b_{k+1}^{k+1} \cdot a_{i,k+1}^k, & i \neq k+1 \end{cases}. \quad (2.7)$$

Цей метод полегшує одержання розв'язку, але супроводжується збільшенням обсягу обчислень.

2.1.1.3 Модифікований метод Гаусса

У багатьох випадках виникає необхідність розв'язання систем лінійних рівнянь із матрицею коефіцієнтів, яка змінюється, і постійним стовпцем вільних членів. Найчастіше для розв'язання таких задач використовується модифікований метод Гаусса. В цьому методі матрицю коефіцієнтів A з матричного рівняння (2.1) подають у вигляді добутку лівої і правої трикутних матриць

$$L * R = A.$$

Оскільки діагональні елементи однієї з матриць дорівнюють одиниці, їх можна не запам'ятовувати, і обидві матриці зберігати в пам'яті ЕОМ на місці матриці коефіцієнтів.

Алгоритм модифікованого методу Гаусса наведено на рис. 2.2.

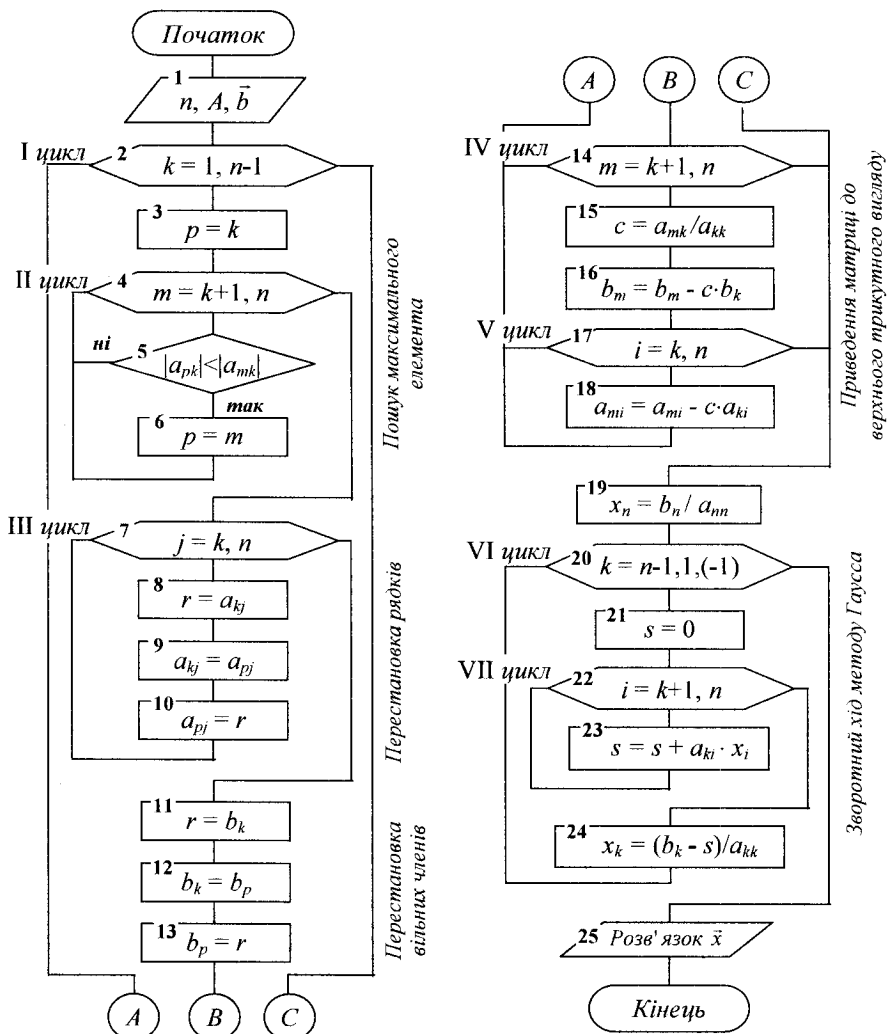


Рисунок 2.2 – Алгоритм модифікованого методу Гаусса

У варіанті методу, що називається **методом Краута**, використовується така послідовність знаходження елементів матриць:

$$K = 1, 2, \dots, n;$$

$$l_{ik} = a_{ik} - \sum_{p=1}^{k-1} l_{ip} r_{pk}, \quad i = k, k+1, \dots, n;$$

$$l_{kk} = \frac{1}{l_{kk}},$$

$$r_{kj} = l_{kk}(a_{kj} - \sum_{p=1}^{k-1} l_{kp}r_{pj}), \quad j = k+1, \dots, n;$$

$$r_{kk} = 1.$$

Система $AX = C$ зводиться до системи $LRX = C$, розв'язання якої замінюється розв'язанням двох систем з трикутними матрицями:

$$LY = C,$$

$$RX = Y.$$

Елементи Y, X знаходять з таких співвідношень:

$$y_1 = l_{11}c_1;$$

$$y_i = l_{ii}(c_i - \sum_{p=1}^{i-1} l_{ip}y_p), \quad i = 2, \dots, n;$$

$$x_i = y_i - \sum_{p=i+1}^n r_{ip}x_p, \quad i = n-1, \dots, 1.$$

Число арифметичних операцій, необхідних для розв'язання цим методом системи лінійних алгебраїчних рівнянь, $N = 2n^2$.

2.1.1.4 Застосування прямого ходу методу Гаусса для пошуку визначників

Для обчислення визначників матриць застосовують два підходи:

- рекурсивний розрахунок за допомогою розкладання за рядком чи стовпцем;
- обчислення на основі прямого ходу методу Гаусса.

Перший спосіб ґрунтується на використанні тієї властивості визначників, що визначник матриці дорівнює сумі добутків елементів будь-якого рядка чи стовпця на їх алгебраїчне доповнення, тобто:

$$\det(A) = \sum_{j=1}^n a_{ij} \cdot A_{ij}, \quad \forall i = \overline{1, n}.$$

Таким чином, обчислення одного визначника n -го порядку зводиться до розрахунку n визначників порядку $n-1$. Реалізується даний спосіб за допомогою рекурсії.

Рекурсивний спосіб зручно застосовувати до рядків чи стовпців, що мають велику кількість нульових елементів. Якщо ж нульових елементів у матриці немає або дуже мало, то застосування цього способу є вкрай неефективним. Для визначника n порядку доведеться розрахувати $n!/2$ визначників другого порядку.

Метод, що базується на алгоритмі прямого ходу методу Гаусса, використовує властивість визначника трикутної матриці. Для такої матриці визначник дорівнює добутку елементів головної діагоналі.

Для обчислення визначника використовується алгоритм побудови послідовності матриць $A \rightarrow A^1 \rightarrow A^2 \rightarrow \dots \rightarrow A^n$ прямого ходу методу Гаусса з тією відмінністю, що при перестановці рядків чи стовпців знак визначника змінюється на протилежний. Значення визначника розраховується за формулою:

$$\det(A) = (-1)^m \cdot a_{11} \cdot a_{22} \cdot \dots \cdot a_{nn}^{n-1},$$

де m – кількість перестановок.

Цей метод дозволяє обчислювати визначники матриць великих порядків.

2.1.1.5 Метод Крамера

Даний метод полягає у розрахунку визначника $\det(A)$ матриці A , а також визначників $\det(A_k)$ матриць A_k , $k = \overline{1, n}$. Матриці A_k отримуються з матриці A шляхом заміни k -го стовпця на стовпець B .

Можливі варіанти.

1. Якщо $\det A \neq 0$, то система має єдиний розв'язок $\bar{x} = (x_1, \dots, x_n)$, який знаходиться за формулою:

$$x_k = \frac{\det(A_k)}{\det(A)}; \quad k = \overline{1, n}.$$

2. Якщо $\det(A) = 0$ та всі $\det(A_k) = 0$, $k = \overline{1, n}$, то розв'язків у системі нескінченна множина.

3. Якщо $\det(A) = 0$ і хоча б один $\det(A_k) \neq 0$, то система розв'язків не має.

Метод Крамера (визначників) не може бути застосований в більшості практичних задач через велику складність розрахунку визначників навіть при невеликому зростанні порядку системи.

Приклад розв'язання СЛАР методом Крамера наведено у Додатку А (2-а частина посібника).

$$\begin{cases} y_0 - u_0 y_1 = v_0, \\ y_1 - u_1 y_2 = v_1, \\ \dots\dots\dots \\ y_{i-1} - u_{i-1} y_i = v_{i-1}, \\ y_i - u_i y_{i+1} = v_i, \\ y_{n-1} - u_{n-1} y_n = v_{n-1}, \\ y_n = v_n, \end{cases}$$

де $u_0, v_0, u_1, v_1, \dots, u_n, v_n$ – деякі коефіцієнти, що називаються прогоночними.

Відзначимо, що

$$u_0 = -\frac{c_0}{b_0}; \quad v_0 = \frac{\Phi_0}{b_0}.$$

Прогоночні коефіцієнти, визначені із останньої системи, дають можливість знайти y_n, y_{n-1}, \dots, y_0 .

Вилучаючи із попередніх співвідношень y_{i-1} шляхом арифметичних перетворень, одержуємо формули для визначення шуканих значень:

$$u_i = -\frac{c_i}{a_i u_{i-1} + b_i},$$

$$v_i = \frac{\Phi_i - a_i v_{i-1}}{a_i u_{i-1} + b_i}$$

і далі

$$y_n = v_n,$$

$$y_i = u_i y_{i+1} + v_i, \quad i = n-1, \dots, 1, 0.$$

2.1.2 Ітераційні методи

Ітераційні методи особливо ефективні при високому порядку системи. Вони застосовуються в системах, попередньо приведених до вигляду:

$$\begin{cases} x_1 = b_{1,n} x_n + b_{1,n-1} x_{n-1} + \dots + b_{1,1} x_1 + b_{1,0}, \\ x_2 = b_{2,n} x_n + b_{2,n-1} x_{n-1} + \dots + b_{2,1} x_1 + b_{2,0}, \\ \dots \\ x_n = b_{n,n} x_n + b_{n,n-1} x_{n-1} + \dots + b_{n,1} x_1 + b_{n,0}. \end{cases} \quad (2.8)$$

або в матричній формі: $X = BX + B_0$,

$$\text{де } B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \dots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix},$$

$$B_0 = \begin{bmatrix} b_{10} \\ b_{20} \\ \vdots \\ b_{n0} \end{bmatrix}.$$

Існує декілька основних різновидів ітераційних методів. Це методи Якобі (простої ітерації), Гаусса-Зейделя і послідовної верхньої релаксації, в основі яких лежить систематичне уточнення значень змінних, заданих на початку розрахунку.

В методі Якобі початкові значення змінних використовуються для обчислення нових значень x_1, x_2, \dots, x_n за допомогою наведених нижче рівнянь. Процес припиняється, коли всі нові значення виявляються достатньо близькими до початкових. В іншому випадку нові значення використовуються замість початкових. Ця процедура повторюється доти, доки не буде досягнута збіжність або стане ясно, що процес розбіжний. В цьому методі заміна значень всіх змінних проводиться одночасно (одночасне зміщення).

Система ітераційних рівнянь має вигляд:

$$\begin{cases} x_1^{(m+1)} = b_{11}x_1^{(m)} + b_{12}x_2^{(m)} + b_{13}x_3^{(m)} + \dots + b_{1n}x_n^{(m)} + b_{10}, \\ x_2^{(m+1)} = b_{21}x_1^{(m)} + b_{22}x_2^{(m)} + b_{23}x_3^{(m)} + \dots + b_{2n}x_n^{(m)} + b_{20}, \\ \vdots \\ x_n^{(m+1)} = b_{n1}x_1^{(m)} + b_{n2}x_2^{(m)} + b_{n3}x_3^{(m)} + \dots + b_{nn}x_n^{(m)} + b_{n0}; \end{cases}$$

де $x_i^{(m+1)}, x_i^{(m)}$, відповідно, значення, x_i ($i = 1, \dots, n$) на наступній $(m+1)$ і попередній (m) ітераціях.

В методі Гаусса-Зейделя уточнене значення x_1 відразу ж використовується для обчислення x_2 . Потім за новими значеннями x_1 і x_2 визначаються x_3 і т. д. Це дозволяє істотно збільшити швидкість збіжності.

В методі послідовної верхньої релаксації нові значення кожної змінної обчислюються як:

$$x_i^{(m+1)} = x_i^{(m)} + \omega(\bar{x}_i^{(m+1)} - x_i^{(m)}),$$

де $\bar{x}_i^{(m+1)}$ – уточнене значення $x_i^{(m)}$ за методом Гаусса-Зейделя; ω – параметр релаксації ($1 \leq \omega \leq 2$).

При $\omega=1$ цей метод подібний до методу Гаусса-Зейделя. Швидкість збіжності залежить від ω .

Одним з головних питань щодо застосування ітераційних методів є збіжність. Для оцінювання збіжності обчислюються норми матриці коефіцієнтів $\|B\|$ з системи (2.2).

Найбільшого поширення набули такі способи оцінювання норм:

$$\text{I норма: } \max_{1 \leq j \leq n} \sum_{i=1}^n |b_{ij}|,$$

$$\text{II норма: } \max_{1 \leq i \leq n} \sum_{j=1}^n |b_{ij}|,$$

$$\text{E-норма (Евклідова): } \sqrt{\sum_{i=1}^n \sum_{j=1}^n b_{ij}^2}.$$

Існує декілька підходів до визначення збіжності за допомогою оцінки норм. В загальному випадку достатньо, щоб хоча б одна з норм матриці B була менша за одиницю

$$\|B\| < 1.$$

В математиці називають таку умову «звичайною» або «сильною». В багатьох випадках збіжність забезпечується і при виконанні так званої «слабкої» ознаки. Наприклад, «слабка» ознака сум для рядків: для всіх сум коефіцієнтів рядків ($i = 1, \dots, n$) виконується співвідношення:

$$\sum_{j=1}^n b_{ij} \leq 1,$$

але є один рядок p для якого

$$\sum_{j=1}^n b_{pj} < 1.$$

Аналогічно визначається «слабка» ознака сум для стовпців.

«Слабка» ознака може використовуватися в тих випадках, коли матриця коефіцієнтів A системи рівнянь (2.1) є нерозкладна, тобто це квадратна матриця A , яку (на відміну від розкладної) не можна привести до вигляду:

$$\begin{pmatrix} A_1 & A_2 \\ 0 & A_3 \end{pmatrix},$$

де A_1, A_3 – квадратні матриці.

Для розкладних матриць система рівнянь (2.1) розпадається на дві системи рівнянь, що розв'язуються послідовно. В більш детальних підручниках, список яких наведено в кінці цієї книги, міститься багато більш детальних доведень і аналіз властивостей та ознак для оцінювання збіжності, але для загальноінженерного підходу до багатьох практичних задач достатньо даних, що наведені вище.

2.2 Визначення власних значень матриць

2.2.1 Постановка проблеми

Багато інженерних задач зводяться до розгляду систем рівнянь, що мають єдиний розв'язок лише в тому разі, якщо відомо значення деякого внутрішнього параметра. Цей особливий параметр називають характеристичним або власним параметром (значенням) системи. При аналізі автоматичних систем власні значення дозволяють визначити критичні параметри керувальних впливів, перевищення яких призводить до втрати системою стійкості. При динамічному аналізі механічних або радіоелектронних систем з самозбудженням власні значення відповідають резонансним частотам коливань, а власні вектори характеризують амплітуди цих коливань.

Формулювання проблеми

Знайти n скалярних значень λ і власних векторів X , що задовольняють матричне рівняння

$$AX = \lambda X. \quad (2.3)$$

Разом з пересічною проблемою власних значень існує так звана узагальнена проблема власних значень (задача для двох матриць A і F):

$$AX = \lambda FX.$$

Тут ми розглянемо тільки ту проблему, що випливає з рівняння (2.3).

Згадаємо деякі означення з теорії матриць:

- квадратна матриця A називається особливою (сингулярною), якщо

$$\det A = 0;$$

- матриця A називається ортогональною, якщо

$$A^T A = EA,$$

тобто $A^{-1} = A^T$, де T – знак транспонування;

- матриці A і B подібні, якщо існує така несингулярна матриця P , що справедливо співвідношення

$$B = P^{-1}AP.$$

Доцільно перед аналізом методів визначення власних значень зупинитись на властивостях власних значень і векторів:

1. Всі n власних значень симетричної матриці розміру $n \times n$, що складається з дійсних чисел, дійсні;

2. Якщо власні значення матриці різні, то її власні вектори ортогональні. Сукупність n лінійно незалежних власних векторів складає базис простору, що розглядається. Тому для сукупності лінійно незалежних векторів x_i ($i=1,2,\dots,n$) будь-який вектор в тому ж просторі можна виразити через власні вектори;

3. Якщо дві матриці подібні, то їх власні значення збігаються. З подібності A і B виходить, що $B = P^{-1}AP$, оскільки $AX = \lambda X \Rightarrow P^{-1}AX = \lambda P^{-1}X$. Якщо прийняти $X = PY$, тоді $P^{-1}APY = \lambda Y$, тобто A і B не тільки мають однакові λ , але й їх власні вектори зв'язані співвідношенням $X = PY$;

4. Добуток власного вектора матриці на скаляр є власним вектором тієї ж матриці. Як правило, власні вектори нормалізують, розділивши їх на найбільший елемент або на суму квадратів елементів.

2.2.2 Методи обчислення власних значень

Вибір найбільш ефективного методу визначення власних значень або власних векторів для заданої задачі залежить від ряду факторів (тип рівнянь, число і характер власних значень, вид матриці і т. д.). Алгоритми визначення власних значень можна розподілити на 3 групи:

- прями, основані на розкритті характеристичних (вікових) визначників $\det(A - \lambda E) = 0$ і розв'язанні характеристичних рівнянь;
- ітераційні, основані на багатократному застосуванні ітераційного алгоритму, який наближає власний вектор, що одержується в кожному циклі, до точного розв'язку;

- перетворення подібності, які використовують властивості подібних матриць, що мають однакові власні значення і ортогональні власні вектори.

2.2.2.1 Прямі методи

Найбільш очевидним шляхом розв'язання задач на власні значення є їх визначення з системи рівнянь, яка має ненульовий розв'язок лише в разі, коли $\det(A - \lambda E) = 0$.

Розв'язання в цьому випадку складається з двох етапів:

- розгорнення вікового визначника безпосередньо або одним з відомих методів: Данілевського, Крилова, Левер'є, невизначених коефіцієнтів, інтерполювання і т. д.;
- розв'язання одержаного характеристичного рівняння, корені якого і будуть власними значеннями матриці. Для визначення коренів можна використати один з методів розв'язання нелінійних алгебраїчних рівнянь (розділ 3).

В таблиці 2.1 наведені результати порівняння ефективності різних методів розгорнення вікових визначників, де критерієм є кількість обчислювальних операцій, а в таблиці 2.2 – порівняння методів розв'язання з точки зору складності, точності та швидкості збігання. Більш детальний опис цих методів можна знайти в підручниках, список яких наведено в кінці книги.

Таблиця 2.1 – Кількість дій, що використовуються різними методами розгорнення вікового визначника в залежності від його порядку.

| Метод | Порядок матриці | | | | | | | | | |
|-----------------------------|------------------------|----------------------------|-----|-----|-----|-----|-------|-------|--------|--------|
| | 3 | | 4 | | 5 | | 7 | | 9 | |
| | Множення-ділення (М-Д) | Додавання-віднімання (Д-В) | М-Д | Д-В | М-Д | Д-В | М-Д | Д-В | М-Д | Д-В |
| Безпосереднього розгорнення | 12 | 10 | 60 | 46 | 320 | 238 | 13692 | 10078 | 986400 | 725758 |
| Данілевського | 14 | 12 | 42 | 36 | 92 | 80 | 282 | 252 | 632 | 576 |
| Крилова | 67 | 38 | 179 | 118 | 389 | 280 | 1287 | 1022 | 3209 | 2688 |
| Левер'є | 41 | 27 | 153 | 114 | 414 | 330 | 1791 | 1533 | 5228 | 4644 |
| Невизначених коефіцієнтів | 67 | 41 | 171 | 116 | 364 | 265 | 1189 | 945 | 2966 | 2481 |
| Інтерполяції | 46 | 38 | 125 | 102 | 279 | 230 | 972 | 826 | 2525 | 2202 |

Таблиця 2.2 – Вибір алгоритму розв’язання задачі на власні значення

| Назва алгоритму | Застосовується для матриць | Результат | Рекомендується для пошуку власних значень | | | Примітка |
|----------------------------------|----------------------------|----------------------------------|---|-------------|----------------|---|
| | | | Max або min | Всі $x < 6$ | Всі $x \geq 6$ | |
| Визначник (ітерація) | Загального вигляду | Власні значення | | x | | Потребує знаходження коренів полінома загального вигляду |
| Ітерація (ітерація) | Загального вигляду | Власні значення і власні вектори | x | x | x | Забезпечує найкращу точність для найбільшого і найменшого власних значень |
| Метод Якобі (перетворення) | Симетричних | Діагональна форма матриці | | x | x | Теоретично потребує нескінченного числа кроків |
| Метод Гівенса (перетворення) | Симетричних | Тридіагональна форма матриці | | x | x | Потребує знання коренів простого полінома |
| | Несиметричних | Форма Гессенберга | | x | x | Потребує застосування додаткового методу |
| Метод Хаусхолдера (перетворення) | Симетричних | Тридіагональна форма матриці | | x | x | Потребує знання коренів простого полінома |
| Метод LR (перетворення) | Загального вигляду | Квазидіагональна форма матриці | | x | x | Буває нестійкий |
| Метод QR (перетворення) | Загального вигляду | Квазидіагональна форма матриці | | x | x | Найкращий метод, що має найбільшу узагальненість |

2.2.2.2 Ітераційні методи

Ітераційні методи (Iteration methods) основані на багатократному використанні ітераційного алгоритму, що наближає власний вектор, який одержується в кожному циклі, до точного розв’язку.

Розрахунок починається з початкового нормованого вектора $X^{(0)}$, який множиться зліва на матрицю A , і результат дорівнює добутку постійної (власне значення λ) і нормованого вектора $X^{(1)}$:

$$AX^{(0)} = \lambda X^{(1)}.$$

Якщо вектор $X^{(1)}$ збігається з $X^{(0)}$, то розрахунок припиняється. Інакше вектор $X^{(1)}$ використовується як початковий, і всі обчислення повторюються доти, доки $|X_i^{(n)} - X_i^{(n-1)}| \leq \epsilon$, де $X^{(n)} \in \{X_i^{(n)}\}$ і $X^{(n-1)} \in \{X_i^{(n-1)}\}$ – вектори, що одержані в $n-1$ і n ітераційних циклах, а ϵ – задана припустима похибка обчислень.

Якщо процес збігається, одержаний постійний множник відповідає найбільшому власному значенню λ_{max} , а нормований вектор – відповідному власному вектору.

Аналогічно можна знайти найменше власне значення. Для цього початкова система рівнянь попередньо множиться на обернену матрицю A^{-1} :

$$A^{-1}AX = \lambda A^{-1}X,$$

звідки одержуємо $\frac{1}{\lambda} X = A^{-1}X$.

Подальше розв'язання задачі на власні значення за алгоритмом рисунка 2.1 приводить до максимального власного значення $\lambda_{max}^* = \frac{1}{\lambda_{min}}$, за яким знаходиться найменше власне значення λ_{min} .

Визначивши максимальне або мінімальне власне значення, можна знайти наступні за ним за величиною, замінивши вихідну матрицю матрицею, яка містить лише решту власних значень.

Принцип ортогональності власних векторів $X_i^T X_j = 0$ при $i \neq j$ і $X_i^T X_j = 1$ при $i = j$ дозволяє довести твердження, що якщо створити нову матрицю $A^* = A - \lambda_1 X_1 X_1^T$, то вона буде мати власне значення $\lambda_1 = 0$, а всі решта її власних значень будуть збігатися з власними значеннями вихідної матриці A . В результаті знаходимо λ_2 і далі, утворюючи A^{**} і т. д., аналогічно визначаємо всі власні значення і вектори.

2.2.2.3 Методи перетворень подібності

Методи перетворень подібності ґрунтуються на властивостях подібних матриць, які мають однакові власні значення і ортогональні власні вектори.

Ці методи використовуються для одержання з вихідної матриці нової, подібної до неї, але простішого вигляду. Очевидно, найкращим

спрощенням було б приведення матриці до діагонального (або блоково-діагонального) вигляду. В цьому випадку власні значення відповідали б елементам матриці, що стоять на головній діагоналі (в випадку дійсних власних значень), або просто визначались би з квадратних матриць розміру 2×2 (в випадку комплексних), що стоять на головній діагоналі. Приведення до діагонального вигляду часто важко здійснити, тому в результаті більшість методів приводить матрицю до тридіагонального вигляду, для якого існують достатньо добре досліджені і відпрацьовані на практиці методи знаходження власних значень. Як приклад можна навести метод, що приводить до послідовності Штурма і має таку властивість, що корені полінома i -го порядку розташовуються між коренями поліномів $(i+1)$ -го порядку (це значно спрощує їх визначення, оскільки обмежує ділянки відокремлення коренів). Вихідний визначник тридіагональної матриці

$$\det(A - \lambda E) = \begin{vmatrix} a_1 - \lambda & b_2 & & & & \\ & b_2 & a_2 - \lambda & & & \\ & & & \dots & & \\ & & & & a_{n-1} - \lambda & b_n \\ & 0 & & & b_n & a_n - \lambda \end{vmatrix}$$

може бути поданий послідовністю поліномів Штурма

$$f_m(\lambda) = (a_m - \lambda)f_{m-1}(\lambda) - b_m^2 f_{m-2}(\lambda),$$

за умови $f_0(\lambda) = 1$ і $f_1(\lambda) = a_1 - \lambda$.

Приймаючи $m = 2, \dots, n$ і послідовно визначаючи корені, знаходимо власні значення як корені полінома n порядку.

Якщо в разі симетричних матриць ефективним способом знаходження власних значень є приведення матриці до тридіагонального вигляду, то для симетричних матриць доцільно користуватися методами перетворення подібності для приведення вихідної матриці до вигляду Гессенберга:

$$\begin{vmatrix} * & * & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 \\ * & * & * & * & 0 & 0 \\ * & * & * & * & * & 0 \\ * & * & * & * & * & * \end{vmatrix}.$$

Для приведення матриць до тридіагонального вигляду (або вигляду Гессенберга) існує ряд методів, найбільш відомими з яких є методи Якобі, Гівенса, Хаусхолдера, які докладно розглядаються в спеціальній літературі. Переваги має (з точки зору швидкодії і кількості обчислювальних операцій) метод відображень, що запропонований Хаусхолдером. Цей метод дозволяє при виконанні одного кроку перетворень перетворити в нуль одразу всі елементи цілого рядка і стовпця, що стоять зовні трьох діагоналей, в той час як в методі Гівенса на кожному k -му кроці для цього необхідно зробити $(n^2 - 3n + 2)/2$ перетворень, а в методі Якобі створення нового нульового елемента може супроводжуватись його зникненням в іншому місці матриці, тобто не гарантовано одержання результату за кінцеву кількість кроків.

Найбільш загальними з методів перетворень подібності є методи LR і QR , що ґрунтуються на поданні матриці A в вигляді добутків $A = LR$ і $A = QR$, де L – ліва трикутна матриця з одиничними діагональними елементами; R – права трикутна матриця; Q – ортогональна матриця.

Далі, застосовуючи перетворення подібності

$$A^* = L^{-1}AL = L^{-1}(LR)L = RL,$$

$$A^* = Q^TAL = Q^T(QR)Q = RQ,$$

одержуємо послідовність матриць, яка прямує до квазідіагонального вигляду.

2.2.3 Порівняння методів визначення власних значень

Основними критеріями, що дозволяють оцінити ефективність методів знаходження власних значень стосовно їх алгоритмізації і використання для розв'язання задач на ЕОМ, є точність, швидкодія (кількість обчислювальних операцій) і необхідний розмір пам'яті ЕОМ.

Швидкодія методу залежить від двох складових: кількості операцій в єдиному циклі і кількості циклів, необхідних для одержання результату.

Важливим критерієм оцінювання методів визначення власних значень є їх загальність (універсальність), тобто пристосованість до матриць різних видів, незалежно від симетрії і вигляду елементів (дійсні або комплексні).

Проаналізуємо з точки зору цих критеріїв методи знаходження власних значень.

Прямі методи приводять до необхідності розв'язання нелінійних алгебраїчних рівнянь. Методи їх розв'язання достатньо добре розроблені. Але в задачах на власні значення часто зустрічаються кратні корені, при цьому ітераційні методи не гарантують одержання розв'язку. Швидкодія цих методів різко зменшується при підвищенні порядку матриці внаслідок

розкриття вікового визначника.

Ці недоліки ускладнюють використання прямих методів знаходження власних значень для великих матриць (порядку більше 10).

Істотним недоліком ітераційних методів розв'язання задач знаходження власних значень є накопичення помилок при кожному кроці ітераційного процесу (при визначенні максимального значення, створенні нової матриці і т. д.), тому практично ними можна користуватися для визначення не більше трьох-чотирьох власних значень.

В інших випадках доцільно користуватись методами перетворення подібності.

Найбільш ефективні методи перетворення подібності гарантують одержання результату при достатньо високій точності розв'язання. Часто найефективніше розв'язання задачі забезпечується при одночасному використанні декількох методів перетворення подібності. Наприклад, методом Хаусхолдера матриця приводиться до вигляду Гессенберга, а потім методом QR визначаються власні значення.

Таким чином, вибір найбільш зручного алгоритму для розв'язання різних задач знаходження власних значень визначається типом власних значень, виглядом матриці і кількістю шуканих власних значень. Чим більш складна задача, тим менше методів та алгоритмів, з яких можна вибирати. Найбільш універсальний алгоритм QR , але він і один з найскладніших. При малих порядках матриці $n < 3...5$ можна користуватись найпростішими прямими та ітераційними методами, а при збільшенні порядку і ускладненні вигляду матриці рекомендуються лише методи перетворення подібності. Таблиця 2.2 дозволяє полегшити вибір методу розв'язання задачі на власні значення. Звичайно пакети математичного забезпечення ЕОМ містять програми, в яких використовується більшість з цих алгоритмів. Одним з ефективних засобів використання наявних ресурсів ЕОМ є одночасне застосування двох підпрограм, що дозволяє поєднати їх найкращі якості.

Контрольні запитання та завдання

1. Чим відрізняються прямі та непрямі методи розв'язання систем лінійних рівнянь? Дати порівняльну оцінку.
2. Розкрити суть методів Крамера, Гаусса (і його різновидів), прогонки.
3. Як подається система рівнянь для застосування ітераційних методів?
4. Як перевірити збіжність ітераційних алгоритмів?
5. Чим відрізняються алгоритми ітераційних методів Якобі, Гаусса-Зейделя, послідовної верхньої релаксації?
6. Розв'язати наведену нижче систему рівнянь методом Крамера. Скласти алгоритм і програму розв'язання.

$$\begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 = 15; \\ x_1 - x_3 + 7x_4 = 26; \\ x_1 - 2x_2 + 3x_3 - 4x_4 = -10; \\ x_1 - x_2 + 2x_3 + 3x_5 = 20; \\ x_1 + x_3 - x_4 + 10x_5 = 50. \end{cases}$$

7. Розв'язати систему рівнянь з прикладу 6 методом Гаусса. Скласти алгоритм і програму.

8. Розв'язати систему рівнянь з прикладу 6 методом Гаусса-Жордана. Скласти алгоритм і програму.

9. Розв'язати систему рівнянь з прикладу 6 модифікованим методом Гаусса. Скласти алгоритм і програму.

10. Чи можна розв'язати систему рівнянь з прикладу 6 методом прогонки? Скласти алгоритм і програму для розв'язання методом прогонки системи з n рівнянь та тридіагональною матрицею коефіцієнтів.

11. Розв'язати вказаними в п. 6, 7, 8, 9, 10 методами на ЕОМ таку систему:

$$11.1 \quad A = \begin{vmatrix} 1 & 0 & 1 & 2 \\ 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 2 \\ -1 & -1 & -2 & 0 \end{vmatrix} \quad B = \begin{vmatrix} 12 \\ 30 \\ 14 \\ 3 \end{vmatrix}$$

$$11.2 \quad A = \begin{vmatrix} 2 & 0 & 7 & 1 \\ 1 & 2 & -1 & -1 \\ -1 & -2 & 1 & 2 \\ 0 & 2 & 0 & 1 \end{vmatrix} \quad B = \begin{vmatrix} 11 \\ 2 \\ 0 \\ 6 \end{vmatrix}$$

$$11.3 \quad A = \begin{vmatrix} 1 & 3 & 4 & 5 & 1 \\ 1 & 1 & 2 & -1 & 3 \\ -3 & -2 & 1 & 2 & 0 \\ 1 & 1 & 1 & 2 & 1 \\ -1 & 1 & 2 & 3 & 15 \end{vmatrix} \quad B = \begin{vmatrix} 38 \\ 5 \\ 4 \\ 14 \\ 19 \end{vmatrix}$$

12. Розв'язати методом Якобі наведену нижче систему. Скласти алгоритм і програму. Перевірити збіжність.

$$\begin{cases} x_1 = \frac{x_2}{2} + \frac{x_3}{3} - \frac{x_4}{4}, \\ x_2 = \frac{x_1}{10} + \frac{x_2}{10} - \frac{x_3}{10} + \frac{x_4}{2}, \\ x_3 = \frac{x_1}{5} + \frac{x_2}{5} + \frac{x_3}{3} + \frac{x_4}{10}, \\ x_4 = \frac{x_3}{3} + \frac{x_4}{4} + 2. \end{cases}$$

13. Розв'язати методом Гаусса-Зейделя систему з п. 12. Скласти алгоритм і програму. Перевірити збіжність
14. Розв'язати методом послідовної верхньої релаксації систему з п. 12 з параметром релаксації $\omega = 1,5$.
15. Сформулювати проблему визначення власних значень матриці.
16. Яка матриця називається несингулярною?
17. Сформулювати умову ортогональності матриці.
18. Назвати властивості власних значень та векторів.
19. Чим відрізняються прямі та ітераційні методи визначення власних значень та векторів?
20. Визначити прямим методом власні значення та вектори матриць:

$$A = \begin{vmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \\ 3 & 3 & 1 \end{vmatrix}; \quad A = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 2 & 0 \\ 2 & 1 & 2 \end{vmatrix}$$

21. Скласти алгоритм, програму та визначити власні значення матриць з п. 20 ітераційним методом.
22. Дати загальну характеристику методів перетворень подібності.
23. Чим відрізняються LQ та QR методи ?
24. Дати порівняльну оцінку, аналіз ефективності та умов застосування різних методів визначення власних значень та векторів.

РОЗДІЛ 3 НЕЛІНІЙНІ ЗАДАЧІ

В цьому розділі розглянуті методи розв'язання нелінійних рівнянь і систем рівнянь. До них зводиться багато практичних задач, наприклад, розрахунки нелінійних електричних кіл та систем керування, розв'язання нелінійних диференціальних рівнянь, аналіз стійкості систем шляхом оцінювання їх власних значень та ін.

Якщо для найпростіших видів алгебраїчних рівнянь (не вище третього степеня) існують точні аналітичні формули, то для трансцендентних рівнянь (transcendental equation) і будь-яких систем рівнянь таких методів взагалі не існує і слід користуватися тільки наближеними ітераційними методами та алгоритмами, найбільш поширені з яких розглянуто нижче.

3.1 Розв'язання нелінійних рівнянь

Рівняння, в які входять тільки степені аргументу з відповідними коефіцієнтами, називаються алгебраїчними.

Загальний вигляд алгебраїчного рівняння:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0. \quad (3.1)$$

Нелінійні рівняння, що містять тригонометричні, експоненціальні, логарифмічні або інші спеціальні функції, називають трансцендентними. Наприклад, трансцендентне рівняння може мати вигляд:

$$x^3 + x \sin(x) - 2 = 0.$$

Геометричний сенс розв'язку рівняння $f(x) = 0$, як алгебраїчного, так і трансцендентного, полягає у знаходженні точки перетину графіка функції $f(x)$ з віссю OX .

Можна виділити деякі важливі властивості алгебраїчних рівнянь, що спрощують визначення коренів. Тут і далі ми називаємо деякі властивості теоремами, як це прийнято в математиці, але наводимо їх без доведень, щоб не ускладнювати сприйняття матеріалу. За власним бажанням можна довести ці теореми як додаткові самостійні завдання. З метою полегшення розуміння матеріалу нагадуємо необхідні означення.

Функція $f(x)$ *неперервна в точці* x_0 , якщо $\lim_{x \rightarrow x_0} f(x) = f(x_0)$. Функція $f(x)$ *неперервна на відрізку* $[a, b]$, якщо вона неперервна в будь-якій точці цього відрізка.

Похідною функції $f(x)$ в точці x_0 називають:

$$\lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = f'(x_0).$$

Якщо функція має похідну в точці, то вона неперервна в ній. Обернене твердження не є достовірним.

Функція $f(x)$ **монотонна на відрізку** $[a, b]$, якщо її похідна має однаковий знак (>0 або <0) на всьому відрізку $[a, b]$.

Функція $f(x)$ **опукла (ввігнута)** в точці x_0 , якщо її друга похідна $f''(x_0) > (<) 0$.

1. **ОСНОВНА ТЕОРЕМА АЛГЕБРИ.** Алгебраїчне рівняння порядку n має n коренів, які можуть бути як дійсними, так і комплексними.

Кожен корінь рахується відповідну кількість разів, що дорівнює його кратності. Кратність кореня x_0 дорівнює k , якщо

$$f'(x_0) = f''(x_0) = \dots = f^{(k-1)}(x_0) = 0.$$

2. Якщо всі коефіцієнти a_i рівняння (3.1) дійсні, то всі комплексні корені утворюють комплексно-спряжені пари.

3. **ТЕОРЕМА ДЕКАРТА.** Число позитивних дійсних коренів дорівнює (або менше) числу змін знаків в послідовності коефіцієнтів (те ж твердження справедливе відносно числа від'ємних дійсних коренів при заміні в (3.1) x на $-x$).

4. **ТЕОРЕМА ЛАГРАНЖА.** Верхня межа додатних дійсних коренів визначається як

$$R = 1 + \sqrt[k]{\frac{B}{a_0}}, \quad a_0 > 0,$$

де k – номер першого від'ємного коефіцієнта; B – найбільша абсолютна величина від'ємного коефіцієнта.

5. **ТЕОРЕМА ГЮА.** Якщо рівняння (3.1) має дійсні корені і дійсні коефіцієнти, то

$$a_k^2 > a_{k-1}a_{k+1}.$$

Нагадаємо, що прямі аналітичні методи існують лише для алгебраїчних рівнянь не вище третього порядку, а для трансцендентних рівнянь прямих методів взагалі не існує. При визначенні дійсних коренів чисельними методами треба враховувати три теореми. Перші дві дозволяють відокремити корені, тобто встановити якомога тісніші інтервали $[a, b]$, в яких знаходиться один і тільки один корінь рівняння, а третя – оцінити ступінь наближення.

ТЕОРЕМА 1. Якщо неперервна функція $f(x)$ приймає значення різних знаків на кінцях відрізка $[a, b]$, тобто $f(a)f(b) < 0$, то всередині цього відрізка міститься щонайменше один корінь рівняння $f(x) = 0$, тобто знайдеться хоча б одне число $\xi \in (a, b)$ таке, що $f(\xi) = 0$.

ТЕОРЕМА 2. Якщо монотонна функція $f(x)$ приймає значення різних знаків на кінцях відрізка $[a, b]$, тобто $f(a)f(b) < 0$, то на цьому відрізку міститься рівно один корінь рівняння $f(x) = 0$.

ТЕОРЕМА 3. Нехай ξ – точний, а \bar{x} – наближений корені рівняння $f(x) = 0$, які знаходяться на одному й тому ж відрізку $[a, b]$, причому $|f'(x)| \geq m$, $a \leq x \leq b$.

Тоді

$$|\bar{x} - \xi| \leq \frac{|f(\bar{x})|}{m}.$$

Розв'язання нелінійних рівнянь складається з двох етапів.

На першому етапі необхідно знайти відрізок $[a, b]$, на якому функція має рівно один нуль (відділення коренів).

На другому етапі відбувається уточнення кореня на знайденому відрізку за допомогою одного з чисельних методів з заданою точністю.

Відділити корінь x^* рівняння $f(x) = 0$ – значить вказати окіл точки x^* , який не містить інших коренів цього рівняння.

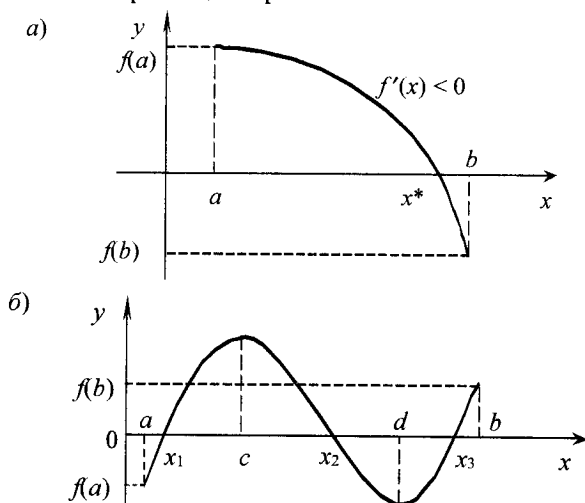


Рисунок 3.1 – Відділення коренів рівняння

Якщо неперервна функція $f(x)$ на кінцях відрізка $[a, b]$ приймає значення різних знаків, тобто якщо $f(a)f(b) < 0$, то всередині цього відрізка існує, принаймні, один корінь рівняння $f(x) = 0$ (див. рис. 3.1). При цьому корінь x^* буде єдиним, якщо $f'(x)$ зберігає знак усередині інтервалу $[a, b]$ (див. рис. 3.1, а).

На практиці відділення кореня рівняння $f(x) = 0$ на відрізку $[a, b]$ починається з перевірки умови $f(a)f(b) < 0$. Якщо ця умова виконана, то на $[a, b]$ є корінь і подальше завдання полягає в з'ясуванні його єдиничності або неєдиничності.

Для відділення коренів практично досить провести процес половинного ділення, відповідно до якого відрізок $[a, b]$ ділиться на 2, 4, 8, ... рівних частин і послідовно визначаються знаки функції в точках поділу. При цьому якщо в точках поділу x_i, x_{i+1} виконується умова $f(x_i) \cdot f(x_{i+1}) < 0$, то на інтервалі (x_i, x_{i+1}) є корінь рівняння $f(x) = 0$. При визначенні коренів завжди намагаються знайти інтервал (x_i, x_{i+1}) якомога меншої довжини.

Графічне відділення коренів рівняння полягає в знаходженні абсциси точки перетину графіка $y = f(x)$ з прямою $y = 0$, тобто віссю абсцис. При цьому, якщо побудувати графік $y = f(x)$ складно, то функцію подають в еквівалентному вигляді

$$f_1(x) = f_2(x)$$

з таким розрахунком, щоб графіки $y_1 = f_1(x)$ і $y_2 = f_2(x)$ було простіше побудувати. Абсциси їх точок перетину будуть коренями рівняння. При графічному відділенні коренів рівняння результат залежить від точності побудови графіків.

Існує ряд методів наближеного розв'язання нелінійних рівнянь, доцільність застосування кожного із яких визначається видом рівняння, його порядком, потрібною точністю і т. д. При розгляді методів будемо вважати, що корені вже відділені.

Слід також пам'ятати, що при визначенні великої кількості коренів знижувати степінь початкового нелінійного рівняння шляхом ділення на $(x - x_i)$ (де x_i – знайдений корінь) треба дуже обережно, що пов'язано з накопиченням похибки розповсюдження, яка буде в коефіцієнтах нового рівняння.

3.1.1 Метод половинного ділення

Нехай дано рівняння $f(x)=0$. Необхідно знайти його корінь з точністю ε на відрізку $[a, b]$, на якому функція неперервна і у кінцях має значення різних знаків, тобто $f(a)f(b)<0$. Таким чином, згідно з теоремою 1, на цьому відрізку існує хоча б один розв'язок рівняння.

Знаходиться середина відрізка $[a, b]$ точка c (рис. 3.2). Корінь може опинитись на відрізку $[a, c]$ або на $[c, b]$, чи збігтися з c . В останньому випадку метод припиняє роботу, інакше за допомогою перевірки виконання умов $f(a)f(c)<0$ і $f(c)f(b)<0$ з'ясується, на якій частині відрізка залишився корінь. Далі процедура повторюється для тієї половини відрізка, на якій є корінь, доки відрізок не зменшиться настільки, що його довжина буде менше від заданої похибки.

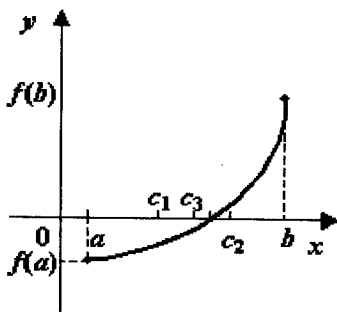


Рисунок 3.2 – Метод половинного ділення

Алгоритм методу

Крок 1. Знаходиться середина відрізка $c := (b+a)/2$.

Крок 2. Перевіряються нижчевикладені умови.

1. Якщо $f(c)=0$ – корінь знайдено.
2. Якщо $f(a)f(c)<0$ – корінь на $[a, c]$, тому $b:=c$.
3. Якщо $f(c)f(b)<0$ – корінь на $[c, b]$, тому $a:=c$.

Крок 3. Перевіряється умова $b-a<\varepsilon$. Якщо вона виконується, то корінь знайдено. В цьому випадку він дорівнює $(a+b)/2$. Інакше повертаються до кроку 1. Блок-схема методу наведена на рис. 3.3.

Похибка розв'язку Δ через n ітерацій знаходиться в межах

$$\Delta \leq \frac{1}{2^n} |x_1 - x_0|.$$

Метод має малу швидкість збіжності, оскільки інтервал, де знаходиться корінь, з кожним кроком зменшується не більше, ніж в два рази.

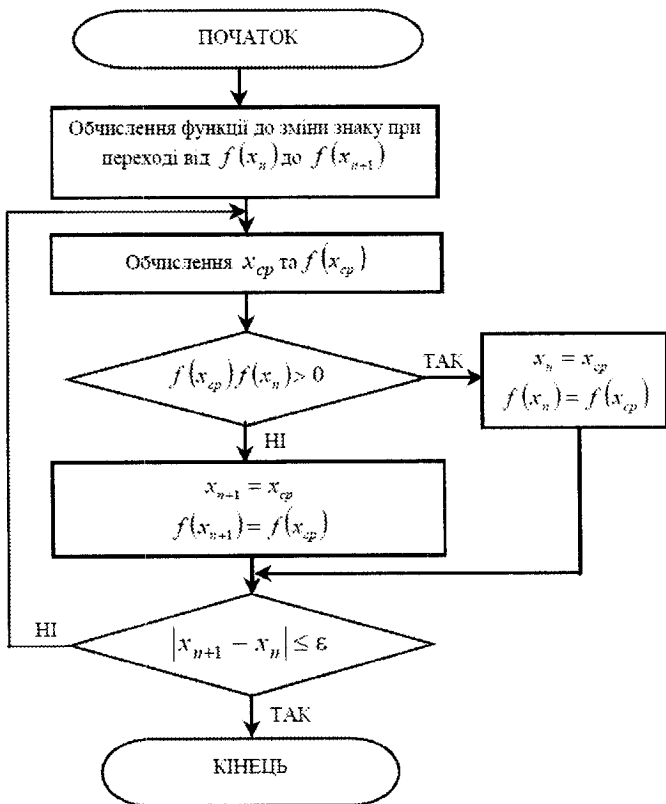


Рисунок 3.3 – Алгоритм методу половинного ділення

3.1.2 Метод хибного положення (хорд)

Нехай дано рівняння $f(x)=0$. Необхідно знайти його корінь з точністю ϵ на відрізку $[a, b]$, на якому функція неперервна і на кінцях має значення різних знаків, тобто $f(a)f(b)<0$. Таким чином, згідно з теоремою 1, на цьому відрізку існує хоча б один розв'язок рівняння.

Ідея методу хорд полягає в заміні на відрізку $[a, b]$ функції $f(x)$ на пряму, що проходить через кінці її графіка (точки А з координатами $(a, f(a))$ та В $(b, f(b))$) (рис. 3.4).

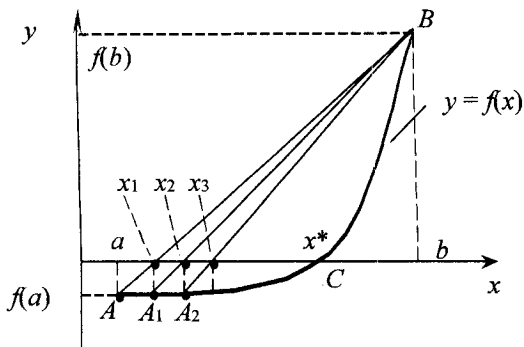


Рисунок 3.4 – Метод хорд

Шуканим коренем C буде перетин $f(x)$ з віссю OX . Рівняння прямої AB запишемо у вигляді:

$$\frac{x-b}{y-f(b)} = \frac{x-a}{y-f(a)}$$

Приймаючи $y = 0$, знаходимо $x_1 = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}$. Цей вираз можна записати в вигляді:

$$\left. \begin{aligned} x_1 &= a - f(a) \frac{b-a}{f(b)-f(a)}; \\ \text{або} \\ x_1 &= b - f(b) \frac{b-a}{f(b)-f(a)}; \end{aligned} \right\} \quad (3.1)$$

Якщо x_1 виявляється недостатньо точним, знаходять друге наближення:

$$x_2 = x_1 - f(x_1) \frac{b-x_1}{f(b)-f(x_1)}. \quad (3.2)$$

На підставі (3.1) і (3.2) можна записати рекурентну послідовність:

$$x_{k+1} = x_k - f(x_k) \frac{b-x_k}{f(b)-f(x_k)}, \quad (3.3)$$

якщо $f(x_k) \cdot f(b) < 0$, і

$$x_{k+1} = x_k - f(x_k) \frac{x_k - a}{f(x_k) - f(a)} \quad (3.4)$$

якщо $f(x_k) \cdot f(a) < 0$.

Алгоритм методу

Крок 1. Знаходиться перша точка $x_1 = a - \frac{f(a) \cdot (b-a)}{f(b) - f(a)}$

Крок 2. Перевіряються нижчевикладені умови.

1. Якщо $f(x_1) = 0$ – корінь знайдено.
2. Якщо $f(a) \cdot f(x_1) < 0$ – корінь на $[a, x_1]$, тому $b := x_1$.
3. Якщо $f(x_1) \cdot f(b) < 0$ – корінь на $[x_1, b]$, тому $a := x_1$.

Крок 3. Запам'ятовується останнє наближення $x_p := x_1$.

Крок 4. Знаходиться нове наближення $x_1 = a - \frac{f(a) \cdot (b-a)}{f(b) - f(a)}$.

Крок 5. Перевіряються нижчевикладені умови.

1. Якщо $f(x_1) = 0$ – корінь знайдено.
2. Якщо $f(a) \cdot f(x_1) < 0$ – корінь на $[a, x_1]$, тому $b := x_1$.
3. Якщо $f(x_1) \cdot f(b) < 0$ – корінь на $[x_1, b]$, тому $a := x_1$.

Крок 6. Перевіряється умова $|x_1 - x_p| < \varepsilon$. Якщо вона виконана, то вважається, що корінь знайдено. В цьому випадку він приймається рівним x_1 . Інакше перехід на крок 3.

Похибка розв'язку оцінюється за формулою:

$$\Delta \leq \frac{M_1 - m_1}{M_1} |x_{n+1} - x_n|,$$

де M_1, m_1 – відповідно, найбільше та найменше значення модуля першої похідної $f(x)$ на відрізку.

Зауваження. Якщо функція $f(x)$ монотонна та опукла (ввігнута), процес наближення до кореня буде відбуватись завжди з одного боку. Тому після визначення на кроці 2 частини відрізка, на якій є корінь, можна запам'ятати її нерухомий кінець і вилучити в циклі перевірку умов на кроці 5.

3.1.3 Метод Ньютона (дотичних)

На відміну від методу хорд, в методі Ньютона функція $f(x)$ замінюється на дотичну і наближення до кореня рівняння визначається точкою перетину дотичної з віссю ОХ.

Рівняння дотичної має вигляд:

$$y = f(x_0) + f'(x_0) \cdot (x - x_0),$$

де x_0 – точка, в якій проведена дотична.

З геометричної точки зору x_{n+1} є значенням абсциси точки перетину дотичної до кривої $y=f(x)$ в точці $(x_n, f(x_n))$ з віссю абсцис. Тому метод Ньютона називають також методом дотичних.

Теорема 1. Якщо $f(x) \in C^2[a, b]$, $f(a)f(b) < 0$, а $f''(x)$ не змінює знака на $[a, b]$, то, виходячи з початкового наближення $x_0 \in [a, b]$, що задовольняє умову $f(x_0)f''(x_0) > 0$, можна обчислити методом Ньютона єдиний корінь x_0 рівняння з будь-яким ступенем точності.

Очевидно, що точка перетину дотичної з віссю ОХ, яка визначає перше наближення до кореня, знаходиться за формулою:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

У точці x_1 проводиться ще одна дотична і знаходиться точка її перетину з віссю ОХ (рис. 3.5).

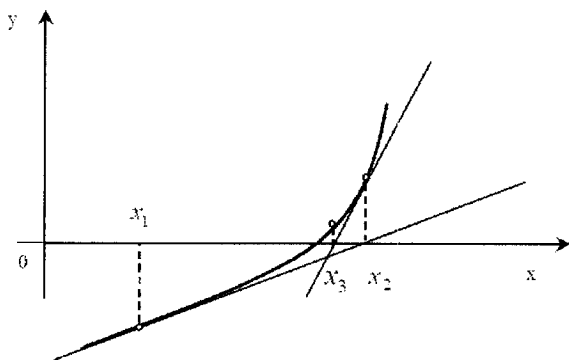


Рисунок 3.5 – Метод Ньютона

Таким чином, наступні наближення розраховуються за формулою:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

В основі цього методу лежить розкладання функції в ряд Тейлора

$$f(x_n + h) = f(x_n) + hf'(x_n) + \frac{h^2}{2} f''(x_n) + \dots$$

Члени, що містять h у другому і більших степенях, відкидаються і в результаті отримується наведена вище наближена формула для оцінювання x_{n+1} .

Метод завершує роботу тоді, коли відстань між двома останніми точками не стане меншою за необхідну точність ϵ .

Для збіжності алгоритму необхідно, щоб функція $f(x)$ була монотонна та опукла (ввігнута) на відрізку $[a, b]$. Коли в процесі обчислень кут нахилу дотичної $f'(x)$ перетворюється на нуль, застосування цього методу ускладнюється. Можна також показати, що у випадку дуже великих значень $f''(x)$ чи кратних коренів метод Ньютона стає неефективним.

Початкове наближення слід вибирати за формулою:

$$x_0 = \begin{cases} a, & f(a) \cdot f''(a) > 0 \\ b, & f(b) \cdot f''(b) > 0 \end{cases}.$$

Як і в методі хорд, при програмній реалізації методу необхідно запам'ятовувати лише дві останні точки наближення.

Алгоритм методу

Крок 1. Знаходиться перша точка $x_1 = \begin{cases} a, & f(a) \cdot f''(a) > 0 \\ b, & f(b) \cdot f''(b) > 0 \end{cases}$.

Крок 2. Запам'ятовується останнє наближення $x_p := x_1$.

Крок 3. Знаходиться нове наближення $x_i = x_p - \frac{f(x_p)}{f'(x_p)}$.

Крок 4. Перевіряється умова $|x_i - x_p| < \epsilon$. Якщо вона виконується, то корінь знайдено. В цьому випадку він приймається рівним x_i . Інакше перехід на крок 2.

Похибка методу оцінюється як

$$\Delta \leq \frac{M_2}{2m_1} (x_{n+1} - x_n)^2,$$

де M_2 – найбільше за модулем значення другої похідної на інтервалі $[x_n, x_{n+1}]$.

3.1.4 Метод січних

Однією з головних проблем при застосуванні методу Ньютона є необхідність аналітичного опису похідної. Якщо це складно чи неможливо, то можна застосувати її наближену оцінку. Тоді замість методу дотичних застосовується метод січних, за яким

$$x_{n+1} = x_n - \frac{f(x_n)}{F'(x_n)},$$

де $F'(x_n)$ – наближена оцінка похідної, що розглядається як січна, а не як дотична, і може бути оцінена за формулою

$$F'(x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}},$$

чи

$$F'(x_n) = \frac{f(x_n + h) - f(x_n)}{h},$$

де h – деякий невеликий крок.

Алгоритм цього методу подібний до алгоритму методу Ньютона, але з іншою ітераційною формулою.

Геометрична інтерпретація методу наведена на рис. 3.6.

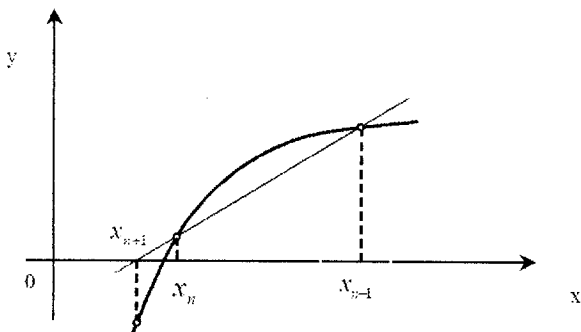


Рисунок 3.6 – Метод січних

3.1.5 Метод простої ітерації

Метод простої ітерації застосовується до розв'язування нелінійного рівняння вигляду

$$x = \varphi(x).$$

Вибравши нульове наближення x_0 , наступні наближення знаходяться за формулою

$$x_{n+1} = \varphi(x_n), \quad n = 0, 1, 2, \dots \quad (3.6)$$

Наведемо достатні умови збіжності методу простої ітерації.

Теорема 2. Нехай для вибраного початкового наближення x_0 на проміжку

$$S = \{x : |x - x_0| \leq \delta\}$$

функція $\varphi(x)$ задовольняє умову Ліпшица

$$|\varphi(x') - \varphi(x'')| \leq q|x' - x''|, \quad x', x'' \in S, \quad (3.7)$$

де $0 < q < 1$, і виконується нерівність

$$|\varphi(x_0) - x_0| \leq (1 - q)\delta.$$

Тоді рівняння $x = \varphi(x)$ має на проміжку S єдиний корінь x_* , до якого збігається послідовність (3.6), причому швидкість збіжності визначається нерівністю

$$|x_n - x_*| \leq \frac{q^n}{1 - q} |\varphi(x_0) - x_0|.$$

Зауваження: якщо функція $\varphi(x)$ має на проміжку S неперервну похідну $\varphi'(x)$, яка задовольняє умову

$$|\varphi'(x)| \leq q < 1,$$

то функція $\varphi(x)$ буде задовольняти умову (3.7) теореми 1.

Наведемо ще одну оцінку, що характеризує збіжність методу простої ітерації:

$$|x_n - x_*| \leq \frac{q}{1-q} |x_n - x_{n-1}|.$$

Похибка методу на n -ій ітерації

$$\Delta \leq \frac{q^n}{1-q} |x_1 - x_0|.$$

Алгоритм методу

Крок 1. Знаходиться перша точка $x_1 := (a + b) / 2$.

Крок 2. Запам'ятовується останнє наближення $x_p := x_1$.

Крок 3. Знаходиться нове наближення $x_t = g(x_p)$.

Крок 4. Перевіряється умова $|x_t - x_p| < \varepsilon$. Якщо вона виконана, то вважається, що корінь знайдено. В цьому випадку він приймається рівним x_t . Інакше перехід на крок 2.

Геометрична інтерпретація методу наведена на рис. 3.7.

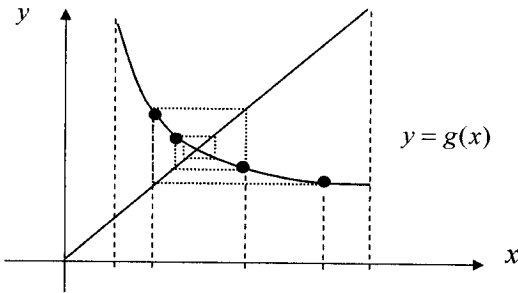


Рисунок 3.7 – Метод простої ітерації для випадку $g'(x) < 0$

3.1.6 Визначення комплексних коренів

Для визначення комплексних коренів (complex roots) можна застосовувати ті ж самі методи, що й для дійсних коренів, але при цьому оперують вже арифметикою комплексних чисел (контроль збіжності та похибки ведеться за модулем комплексного числа), що не завжди зручно для користувача.

Існує низка спеціальних методів, що дозволяють оцінювати комплексні корені, проводячи обчислення з дійсними числами. Більшість з цих методів базується на перетворенні початкового алгебраїчного рівняння (3.1) до добутку квадратичних співмножників типу

$$x^2 + px + q,$$

де p, q – коефіцієнти.

Проміжною формою для здійснення такого перетворення є рівняння у вигляді

$$(x^2 + px + q) \cdot (x^{n-2} + b_{n-1}x^{n-3} + \dots + b_3x + b_2) + b_1x + b_0 = 0, \quad (3.8)$$

де $b_1x + b_0$ – лінійний залишковий член, який прагнуть звести до нуля, і щоб початкове рівняння (3.1) ділилося на квадратичний співмножник $x^2 + px + q$ без залишку.

Для того, щоб знайти коефіцієнти $b_{n-1}, b_{n-2}, \dots, b_3, b_2$ при припущенні $b_1 = b_0 = 0$, розглянемо систему рівнянь, що виходять з еквівалентності рівнянь (3.1) та (3.2):

$$\begin{cases} b_{n-1} = a_{n-1} - p, \\ b_{n-2} = a_{n-2} - pb_{n-1} - q, \\ \vdots \\ b_{n-j} = a_{n-j} - pb_{n+1-j} - qb_{n+2-j}, \\ b_3 = a_3 - pb_4 - qb_5, \\ b_2 = a_2 - pb_3 - qp_4, \\ 0 = a_1 - pb_2 - qb_3, \\ 0 = a_0 - qb_2. \end{cases} \quad (3.9)$$

Вона може бути розв'язана методом прогонки чи ітераційним методом Ліна, алгоритм якого наведений на рисунку 3.8.

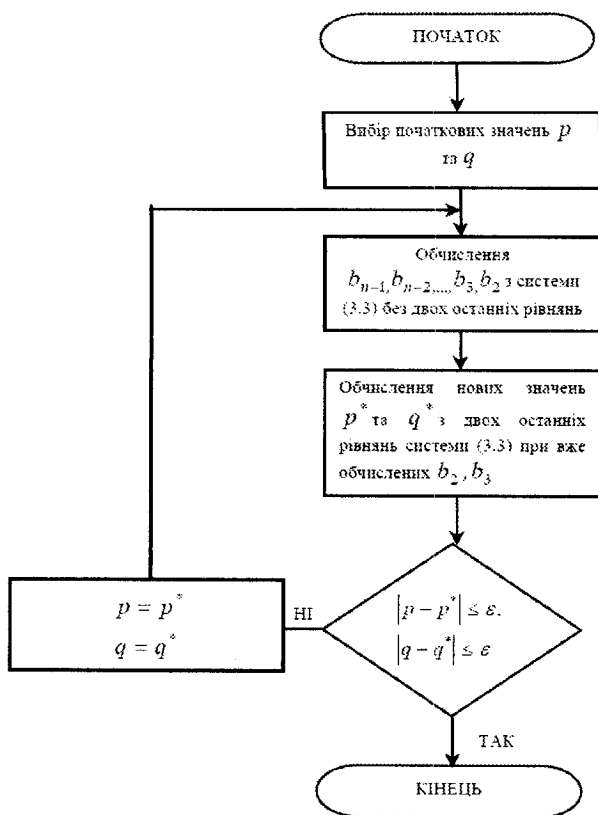


Рисунок 3.8 – Алгоритм методу Ліна

3.2 Розв’язання систем нелінійних рівнянь

В загальному випадку система з n нелінійних рівнянь з n невідомими подається у вигляді:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0, \\ f_2(x_1, x_2, \dots, x_n) = 0, \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0. \end{cases} \quad (3.10)$$

Оскільки нелінійні функції, що входять до системи, неможливо

$$G'(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \dots & \frac{\partial g_1}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial g_n}{\partial x_1} & \frac{\partial g_n}{\partial x_2} & \dots & \frac{\partial g_n}{\partial x_n} \end{bmatrix}.$$

Для розв'язання систем нелінійних рівнянь широке застосування набув більш стійкий метод – метод Ньютона. Він є аналогом методу Ньютона для одного рівняння і базується на розкладанні всіх n рівнянь у ряд Тейлора:

$$\begin{aligned} f_1(x_1 + \Delta x_1, \dots, x_n + \Delta x_n) &= f_1(x_1, \dots, x_n) + \Delta x_1 \frac{\partial f_1}{\partial x_1} + \dots + \Delta x_n \frac{\partial f_1}{\partial x_n} + R_n; \\ \dots & \dots \\ f_n(x_1 + \Delta x_1, \dots, x_n + \Delta x_n) &= f_n(x_1, \dots, x_n) + \Delta x_1 \frac{\partial f_n}{\partial x_1} + \dots + \Delta x_n \frac{\partial f_n}{\partial x_n} + R_n; \end{aligned}$$

де R_n – члени другого та більших порядків, що в подальшому відкидаються.

Задача зводиться до розв'язання системи лінійних рівнянь

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{bmatrix} = \begin{bmatrix} -f_1 \\ -f_2 \\ \vdots \\ -f_n \end{bmatrix}.$$

В цій системі матрицю частинних похідних називають матрицею Якобі і позначають $W(X)$.

Знайдені для певного $(m+1)$ кроку ітерації значення Δx_i використовуються як поправки до попередніх наближень

$$\begin{cases} x_1^{(m+1)} = x_1^{(m)} + \Delta x_1, \\ \dots \\ x_n^{(m+1)} = x_n^{(m)} + \Delta x_n. \end{cases}$$

Загальна ітераційна формула в матричному поданні має вигляд:

$$X^{(m+1)} = X^{(m)} - W^{-1}[X^{(m)}]F[X^{(m)}],$$

де $F[X^{(m)}]$ – вектор-стовпець значень функцій f_1, f_2, \dots, f_n для наближень $X^{(m)}$; $W^{-1}[X^{(m)}]$ – обернена матриця Якобі.

Алгоритм методу Ньютона наведено на рисунку 3.9.

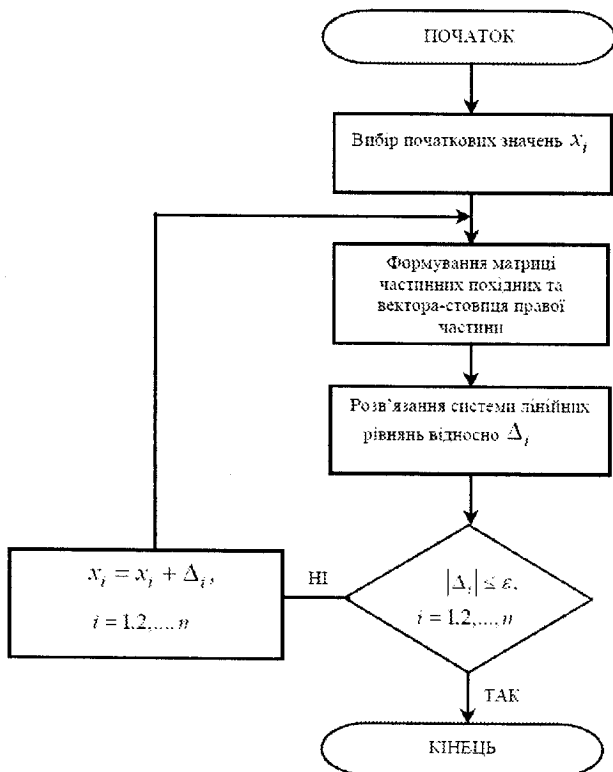


Рисунок 3.9 – Алгоритм методу Ньютона для систем нелінійних рівнянь

Певні труднощі при реалізації алгоритму методу Ньютона виникають при оберненні матриці Якобі. Для цього використовуються відомі з лінійної алгебри способи обернення матриць.

Існує багато варіантів застосування методу Ньютона. Наприклад, модифікований метод Ньютона

$$X^{(m+1)} = X^{(m)} - W^{-1}[X^{(0)}]F[X^{(m)}]$$

В цьому методі не треба обчислювати обернену матрицю Якобі на кожному кроці розрахунків, що спрощує алгоритм, але уповільнює збіжність і робить метод більш чутливим до вибору початкового наближення.

Метод Ньютона з параметром τ

$$X^{(m+1)} = X^{(m)} - \tau W^{-1}[X^{(m)}]F[X^{(m)}]$$

Цей метод дещо схожий з методом послідовної верхньої релаксації для систем лінійних рівнянь.

Застосовуються також різноманітні гібридні методи, в яких поєднується метод Ньютона з методом простої ітерації.

Збіжність методу Ньютона оцінюється шляхом обчислення показника

$$q = \frac{M^2 LP}{2} < 1,$$

де

$$M \geq \|W^{-1}(X)\|,$$

$$L \geq \|W(X)\|,$$

$$P \geq \|F(X)\|,$$

причому

$$\lim_{l \rightarrow \infty} MP \sum_{m=0}^l q^{2^m-1} \rightarrow 0.$$

Похибка на m -ій ітерації визначається нерівністю

$$\Delta \leq MP \frac{q^{2^m-1}}{1 - q^{2^m}}.$$

Контрольні запитання та завдання

1. Навести класифікацію нелінійних рівнянь і систем та методів їх розв'язання. В яких задачах з автоматики та систем управління виникає потреба розв'язання таких задач?

2. Сформулювати головні теореми та правила відносно кількості та виду коренів алгебраїчних рівнянь.

3. Розв'язати алгебраїчне рівняння з комплексними коренями

$$x^6 + 2x^5 + 3x^4 + 4x^3 - x - 1 = 0.$$

Похибка оцінки кожного кореня не повинна перевищувати 0.05.

4. Чому для розв'язання систем нелінійних рівнянь можливе застосування тільки ітераційних методів?

5. Назвати ітераційні методи для розв'язання систем нелінійних рівнянь та порівняти їх.

6. Як оцінюється збіжність ітераційних методів розв'язання систем нелінійних рівнянь?

7. Скласти алгоритм і програму та розв'язати на ЕОМ методами половинного ділення, хорд, Ньютона, січних та простої ітерації (оцінити інтервал збіжності) з похибками $\varepsilon = 0.1, 0.01, 0.001$ такі рівняння:

$$x^4 + x^2 - x + 1 = 0;$$

$$x^3 - 7x + 2 = 0;$$

$$x^5 - 10x^2 + 5 = 0;$$

$$x^3 - 2x^2 - 1 = 0;$$

$$x^5 - x^2 - 3 = 0;$$

$$x^2 - \ln x = 0;$$

$$x - 10 \sin x = 0;$$

$$x^3 - 2 \operatorname{tg} x + 5 = 0;$$

$$x^2 - x + \operatorname{tg} x = 0.$$

Порівняти ефективність (час розрахунків та кількість ітерацій для досягнення заданої похибки) різних методів.

8. Знайти матриці Якобі для систем нелінійних рівнянь:

$$\begin{cases} x_1 + 2x_2 + x_3^2 = 14 \\ \ln|x_1| + x_2^2 - x_3 = 1 \\ x_1 + \sin \frac{\pi x_2}{4} - x_3 = 1 \end{cases}$$

та

$$\begin{cases} \cos x_1 + x_2 + \ln \frac{x_3}{3} = 2 \\ x_1 + x_2^2 + x_3 = 8 \\ \frac{\arctg x_1}{\pi} + \frac{x_2}{2} - \frac{x_3}{12} = 1 \end{cases}$$

9. Як знайти матрицю обернену заданій? Скласти алгоритм.

10. Розв'язати на ЕОМ системи рівнянь з п. 8 методом Ньютона та модифікованим методом Ньютона. Скласти алгоритм та програму. Оцінити збіжність.

11. Дати порівняльну оцінку та рекомендації щодо вибору методів розв'язання нелінійних рівнянь та систем.

РОЗДІЛ 4 МЕТОДИ РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

Переважає більшість об'єктів є нестационарними, вони змінюються у часі під впливом внутрішніх і зовнішніх чинників. Для формального опису нестационарних процесів був розроблений спеціальний математичний апарат, який отримав назву диференціальних рівнянь.

Диференціальними рівняннями (Differential equations) називаються такі рівняння, які крім невідомих функцій однієї або декількох незалежних змінних містять також і їхні похідні. Диференціальні рівняння (ДР) називають звичайними (ЗДР), якщо невідомі функції є функціями однієї змінної, в іншому випадку ДР називаються рівняннями в частинних похідних. Співвідношення

$$F(x, y, y', y'', \dots, y^{(n)}) = 0, \quad (4.1)$$

що поєднує змінну x , невідому функцію $Y = Y(x)$ і її похідні до порядку (n) включно, називають ЗДР n -го порядку.

Рівняння (4.1) часто подають у так званому канонічному вигляді:

$$y^{(n)}(x) = f(x, y(x), y^{(1)}(x), y^{(2)}(x), \dots, y^{(n-1)}(x)).$$

ЗДР має нескінченну множину розв'язків, оскільки функція $y(x)$, що є розв'язком рівняння n -го порядку, містить n невідомих констант, які виникають під час інтегрування. Для пошуку якого-небудь конкретного розв'язку потрібні додаткові умови, які дозволять знайти значення констант. Ці умови можуть бути різними і приводити до різних задач. У випадку, коли додаткові умови задаються при одному значенні незалежної змінної, має місце задача Коші (задача з початковими умовами). Якщо умови задаються для двох або більше значень незалежної змінної, то задача стає крайовою. У задачі Коші додаткові умови називаються початковими, а у крайовій задачі – граничними. Наприклад, для диференціального рівняння n -го порядку початкові умови мають вигляд:

$$\begin{cases} y(a) = y_0 \\ y'(a) = y_1 \\ \dots \\ y^{(n-1)}(a) = y_{n-1} \end{cases},$$

де a – початок відрізка, на якому розв'язується задача; y_0, y_1, \dots, y_{n-1} – деякі числа.

Для практичної реалізації із загального запису ДР (4.1) намагаються виразити старшу похідну, так для $n = 1$ співвідношення (4.1) прийме вигляд:

$$Y' = f(x, Y);$$

$$Y'' = f(x, Y, Y'), \text{ якщо } n = 2.$$

Дамо геометричну інтерпретацію ДР першого порядку. Його розв'язок можна зобразити у вигляді сімейства кривих на площині xOy :

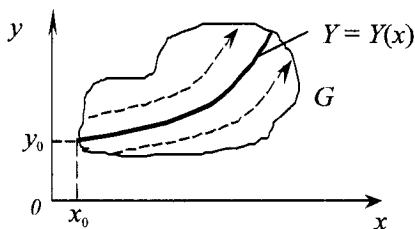


Рисунок 4.1 – Геометрична інтерпретація ДР

Нехай неявна функція $f(x,y)$ визначена та неперервна в області G цієї площини. У кожній точці площини G функція $f(x,y)$ задає деякий напрямок. У цілому це буде поле напрямків. Для загального розв'язку потрібно знайти всі інтегральні криві, дотичні до яких у кожній точці збігаються з напрямком поля. А розв'язок $Y = Y(x)$ є частковим розв'язком, що відповідає певній константі. Через кожен точку з області рішення проходить одна інтегральна крива. Для ДР при $n > 1$ через кожну точку проходить не одна інтегральна крива і потрібно n додаткових умов, тобто для рівнянь вищих порядків геометрична інтерпретація їхніх розв'язків більш складна. А знайти загальний розв'язок в аналітичному вигляді вдається навіть для ДР першого порядку тільки в деяких випадках. Частковий розв'язок теж доводиться шукати наближено.

Методи розв'язання ЗДР можна розбити на такі групи: графічні; аналітичні; наближені аналітичні; чисельні.

Задачу Коші (Cauchy problem) можна сформулювати таким чином.

Нехай дано диференціальне рівняння першого порядку

$$\frac{dy}{dx} = f(x, y). \quad (4.2)$$

Потрібно знайти функцію на відрізьку від $x=a$ до $x=b$, що задовольняє як рівняння (4.1), так і початкову умову $y(a) = y_0$ (при цьому завжди припускається, що існує єдиний розв'язок на всьому відрізьку).

Задача, що полягає в розв'язанні звичайного диференціального рівняння при додаткових умовах, які поставлені при декількох значеннях незалежної змінної, називається крайовою.

Крайову задачу розглянемо на прикладі звичайного диференціального рівняння другого порядку:

$$\frac{d^2 y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right)$$

при граничних умовах $y(a) = A$, $y(b) = B$.

Постановки і методи розв'язання рівнянь більш високих порядків аналогічні.

Для розв'язання диференціальних рівнянь можна використовувати аналітичні й чисельні методи. Аналітичні методи дозволяють знайти розв'язок невеликої кількості типів рівнянь, тому їх застосування є вельми обмеженим. Чисельні методи дозволяють розв'язувати значно більшу кількість задач. В основі застосування чисельних методів лежить дискретизація відрізка $[a, b]$ з кроком h , в результаті чого формується послідовність вузлів $x_k = a + h \cdot k$, $k = \overline{0, n}$, $n = \frac{b-a}{h}$. Питома функція $y(x)$ знаходиться у вигляді таблично заданої функції, значення якої розраховуються у виділених вузлах.

4.1 Методи розв'язання задачі Коші

Основою чисельних методів розв'язання диференціальних рівнянь слугує розкладання функції y в ряд Тейлора в околі початкової точки

$$y(x_0 + h) = y(x_0) + hy'(x_0) + \frac{1}{2} h^2 y''(x_0) + \dots,$$

де h – відстань (крок) між початковою точкою x_0 і точкою $x_1 = x_0 + h$, в якій відшукується розв'язок.

В різних методах враховується різна кількість членів розкладу (в багатокрокових методах в поєднанні з інтерполяційними формулами), що визначає точність обчислень. При використанні цих методів на ЕОМ слід розрізняти похибки округлення через обмеженість кількості значущих цифр в ЕОМ; похибка зрізання (обмеження) – методична похибка, що пов'язана з апроксимацією розв'язків скінченними рядами, замість нескінченних, наприклад, рядами Тейлора.

Внаслідок цих причин виникають два види похибок:

- локальна – сума похибок, що вносяться в обчислювальний процес на конкретному кроці;
 - глобальна (сумарна) – різниця між точним і обчисленим значеннями, яка містить так звану похибку розповсюдження внаслідок накопичення помилок на попередніх етапах обчислення.
- Порядок методу дорівнює p , якщо існує таке позитивне число c , що

$$\Delta \leq ch^{p+1},$$

де Δ – локальна помилка; h – крок дискретизації.

Число c не залежить від номера кроку і його величини, а визначається похідними і довжиною інтервалу. При апроксимації розв'язання рядами Тейлора воно зв'язане зі степенем членів ряду, які відкидаються.

Методи розв'язання задачі Коші поділяють на однокрокові та багатокрокові.

В однокрокових методах для знаходження наступної точки на кривій $y = f(x)$ потрібна інформація лише про один попередній крок (методи Ейлера і Рунге-Кутта).

В багатокрокових методах (прогнозу і корекції) для знаходження наступної точки на кривій $y = f(x)$ потрібна інформація більш ніж про одну з попередніх точок. Щоб отримати достатньо точне чисельне значення часто використовується ітераційна процедура (наприклад, в методах Мілна-Адамса, Башфорта, Хеммінга).

4.1.1 Однокрокові методи

Найбільш простим однокроковим методом, який потребує мінімальних затрат обчислювальних ресурсів, але дає змогу обчислювати результат із порівняно низькою точністю, є метод Ейлера.

В цьому методі для оцінювання наступної точки на кривій $y=f(x)$ використовується лише один лінійний член в формулі Тейлора,

$$y(x_0 + h) = y(x_0) + hy'(x_0),$$

де $y'(x_0)$ визначається з початкового рівняння.

Цей процес можна розповсюдити на наступні кроки:

$$y_{n+1} = y_n + hf(x_n, y_n).$$

Метод Ейлера є методом першого порядку ($p = 1$)

$$\Delta \leq ch^2,$$

де $c = (M_1 + M_0 M_2)/2$, M_0 , M_1 , M_2 – визначаються як

$$M_0 \geq |f(x, y)|,$$

$$M_1 \geq \left| \frac{\partial f(x, y)}{\partial x} \right|,$$

$$M_2 \geq \left| \frac{\partial f(x, y)}{\partial y} \right|$$

для всіх $x \in [a, b]$ і $y = y(x)$.

Метод Ейлера, крім значної похибки зрізання, часто буває нестійким (малі локальні помилки призводять до значного збільшення глобальної).

Цей метод можна вдосконалити різними способами. Найбільш відомі два з них: виправлений метод Ейлера і модифікований метод Ейлера (в літературі зустрічаються інші назви цих методів, наприклад, модифікований метод Ейлера й удосконалений метод ламаних).

Ітераційні формули для цих методів мають вигляд, відповідно:

$$y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_n + h, y_n + hy_n^*)]$$

$$y_{n+1} = y_n + hf \left(x_n + \frac{h}{2}; y_n + \frac{h}{2} y_n^* \right),$$

де $y_n^* = f(x_n, y_n)$.

Геометрична інтерпретація зображена на рисунках 4.2, 4.3.

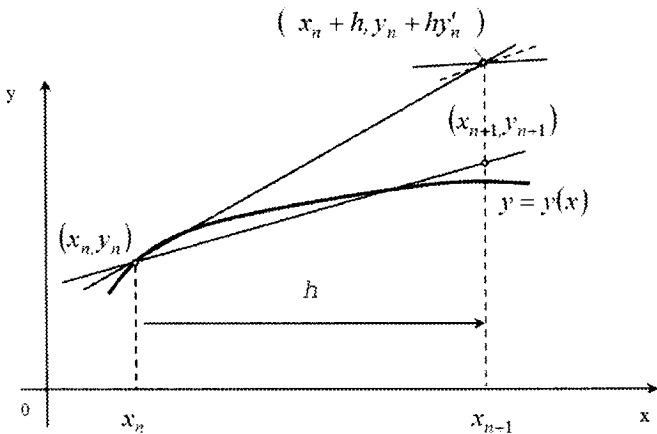


Рисунок 4.2 – Виправлений метод Ейлера

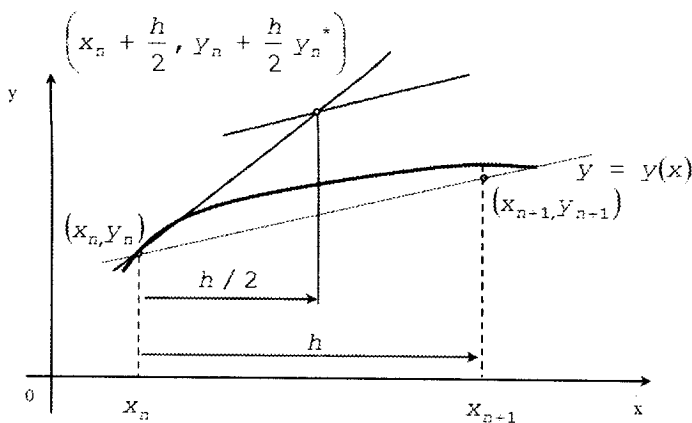


Рисунок 4.3 – Модифікований метод Ейлера

Це методи другого порядку, їх похибка має третій степінь, що досягається покращенням апроксимації похідної. Ідея полягає у спробі зберегти або оцінити член другого порядку у формулі Тейлора. Однак збільшення точності вимагає додаткових витрат машинного часу на обчислення y_n^* . Ще більш висока точність може бути досягнута при обчисленні вищих похідних і збереженні більшої кількості членів ряду Тейлора. Такими методами є методи Рунге-Кутта.

Принцип, на якому побудований модифікований метод Ейлера, можна пояснити, користуючись рядом Тейлора і зберігаючи в ньому член з h^2 . Апроксимація другої похідної $y''(x_0)$ здійснюється кінцевою різницею

$$y''(x_0) = \frac{\Delta y'}{\Delta x} = \frac{y'(x_0 + h) - y'(x_0)}{h}.$$

Аналогічно обчисленню другої похідної в кінцево-різницевому вигляді можна обчислити більш високі похідні: значення n -ї за значеннями попередньої $(n-1)$ -ї.

Метод Рунге-Кутта дає набір формул для обчислення координат внутрішніх точок, які потрібні для реалізації цієї ідеї. Оскільки існує ряд способів знаходження цих точок, то метод Рунге-Кутта об'єднує цілий клас методів для розв'язання диференціальних рівнянь першого порядку.

Найбільш розповсюджений класичний метод четвертого порядку точності:

$$y_{n+1} = y_n + \frac{k_0 + 2k_1 + 2k_2 + k_3}{6},$$

де

$$k_0 = hf(x_n, y_n); k_1 = hf\left(x_n + \frac{h}{2}; y_n + \frac{k_0}{2}\right);$$
$$k_2 = hf\left(x_n + \frac{h}{2}; y_n + \frac{k_1}{2}\right); k_3 = hf(x_n + h; y_n + k_2).$$

Метод Ейлера і його модифікації ще називають методами Рунге-Кутта першого і другого порядку. Метод Рунге-Кутта має значно більш високу точність, що дозволяє збільшити крок розв'язання. Його максимальну величину визначає припустима похибка. Такий вибір часто здійснюється автоматично і входить як складова частина в алгоритм, побудований за методом Рунга-Кутта.

Будь-яку з формул однокрокових методів можна використовувати для розв'язання систем диференціальних рівнянь і диференціальних рівнянь вищих порядків.

Для опису складних динамічних процесів використовують системи диференціальних рівнянь. Найбільше розповсюдження отримали системи диференціальних рівнянь першого порядку:

$$\begin{cases} y_1'(x) = f_1(x, y_1, \dots, y_n) \\ y_2'(x) = f_2(x, y_1, \dots, y_n) \\ \dots \\ y_n'(x) = f_n(x, y_1, \dots, y_n) \end{cases}$$

Для однозначного розв'язку цієї системи задають початкові умови:

$$\begin{cases} y_1(a) = y_1^0 \\ y_2(a) = y_2^0 \\ \dots \\ y_n(a) = y_n^0 \end{cases}$$

Ця система рівнянь з наведеними початковими умовами називається задачею Коші для системи n диференціальних рівнянь першого порядку. У векторному вигляді вона записується так:

$$\begin{cases} \bar{y}'(x) = \bar{f}(x, \bar{y}) \\ \bar{y}(a) = \bar{y}^0 \end{cases}$$

У формулах однокрокових методів для розв'язання такої задачі Коші потрібно лише перейти до векторної форми. Наприклад, формули методу Рунге-Кутта 4-го порядку розв'язання задачі Коші для системи n диференціальних рівнянь першого порядку мають вигляд:

$$y_k(x_{i+1}) = y_k(x_i) + \frac{1}{6} \left(\eta_{1k}^i + 2\eta_{2k}^i + 2\eta_{3k}^i + \eta_{4k}^i \right),$$

де $\eta_{1k}^i = h \cdot f_k(x_i, y_1(x_i), \dots, y_n(x_i))$,

$$\eta_{2k}^i = h \cdot f_k\left(x_i + \frac{h}{2}, y_1(x_i) + \frac{\eta_{11}^i}{2}, \dots, y_n(x_i) + \frac{\eta_{1n}^i}{2}\right),$$

$$\eta_{3k}^i = h \cdot f_k\left(x_i + \frac{h}{2}, y_1(x_i) + \frac{\eta_{21}^i}{2}, \dots, y_n(x_i) + \frac{\eta_{2n}^i}{2}\right)$$

$$\eta_{4k}^i = h \cdot f_k\left(x_i + h, y_1(x_i) + \eta_{31}^i, \dots, y_n(x_i) + \eta_{3n}^i\right)$$

$k = \overline{1, n}$ – номер рівняння,

$i = \overline{0, m}$ – номер вузла, $x_0 = a$.

Задачу Коші для диференціального рівняння порядку n можна звести до задачі Коші для системи n диференціальних рівнянь першого порядку.

Нехай задача Коші має вигляд:

$$\begin{cases} y^{(n)}(x) = f(x, y, y', \dots, y^{(n-1)}) \\ y(a) = y_0 \\ \dots \\ y^{(n-1)}(a) = y_{n-1} \end{cases}$$

Зробимо заміну:

$$\begin{cases} z_1(x) = y'(x) \\ z_2(x) = y^{(2)}(x) \\ \dots \\ z_{n-1}(x) = y^{(n-1)}(x) \end{cases}$$

Як результат отримаємо систему диференціальних рівнянь першого порядку:

$$\begin{cases} y'(x) = z_1(x) \\ z_1'(x) = z_2(x) \\ \dots \\ z_{n-2}'(x) = z_{n-1}(x) \\ z_{n-1}'(x) = f(x, y(x), z_1(x), \dots, z_{n-1}(x)) \end{cases}$$

Початкові умови приймуть вигляд:

$$\begin{cases} y(x_0) = y_0 \\ z_1(x_0) = y_1 \\ \dots \\ z_{n-1}(x_0) = y_{n-1} \end{cases}$$

Як приклад розглянемо розв'язання звичайного диференціального рівняння другого порядку:

$$\frac{d^2 y}{dx^2} = g\left(x, y, \frac{dy}{dx}\right).$$

Нехай $z = \frac{dy}{dx}$, тоді $\frac{dz}{dx} = \frac{d^2 y}{dx^2}$; і система набуває вигляду

$$\begin{cases} \frac{dz}{dx} = g(x, y, z), \\ \frac{dy}{dx} = f(x, y, z), \end{cases}$$

де $f(x, y, z) = z$.

Задача Коші в цьому випадку містить дві початкових умови:

$$y(x_0) = y_0; \quad z(x_0) = y'(x_0) = z_0.$$

Формули Рунге-Кутта для цього випадку мають вигляд:

$$y_{n+1} = y_n + k \quad \text{і} \quad z_{n+1} = z_n + l,$$

$$\text{де } k = \frac{k_0 + 2k_1 + 2k_2 + k_3}{6}, \quad l = \frac{l_0 + 2l_1 + 2l_2 + l_3}{6},$$

$$\begin{aligned}
k_0 &= h z_n, & l_0 &= hg(x_n, y_n, z_n), \\
k_1 &= h \left(z_n + \frac{l_0}{2} \right), & l_1 &= hg \left(x_n + \frac{h}{2}, y_n + \frac{k_0}{2}, z_n + \frac{l_0}{2} \right), \\
k_2 &= h \left(z_n + \frac{l_1}{2} \right), & l_2 &= hg \left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}, z_n + \frac{l_1}{2} \right), \\
k_3 &= h(z_n + l_2), & l_3 &= hg(x_n + h, y_n + k_2, z_n + l_2).
\end{aligned}$$

Раніше було відзначено, що помилка зрізання при використанні методу Рунге-Кутта n -го порядку $\Delta \leq ch^{n+1}$. Обчислення верхніх границь для коефіцієнта c являє собою складну задачу, пов'язану з необхідністю оцінювання ряду додаткових параметрів. Існує декілька способів для оперативного обчислення c . Найбільшого поширення набув екстраполяційний метод Річардсона (ще його називають методом Рунге), коли послідовно знаходять значення y_n з кроком h і з кроком $\frac{h}{2}$, а після цього прирівнюють отримані величини та визначають c з рівняння:

$$y_n^{(h)} + ch^{n+1} = y_n^{\left(\frac{h}{2}\right)} + c \left(\frac{h}{2}\right)^{n+1},$$

що відповідає точному значенню y_n .

Отримаємо оціночне співвідношення:

$$c = \frac{2^{n+1}}{2^{n+1} - 1} \frac{\left[y_n^{\left(\frac{h}{2}\right)} - y_n^{(h)} \right]}{h^{n+1}}.$$

Можна виділити загальні риси однокрокових методів.

1. Щоб отримати інформацію у новій точці, потрібно мати дані лише в одній попередній точці. Цю властивість називають «самостартуванням» (selfstarting).

2. В основі всіх однокрокових методів лежить розкладання функції в ряд Тейлора, в якому зберігаються члени, що містять степені до k включно. Ціле число k називається порядком методу. Похибка на кроці має порядок $k+1$.

3. Всі однокрокові методи не вимагають обчислення похідних — обчислюється лише сама функція, але можуть вимагатися її значення в декількох проміжних точках.

4. Властивість «самостартування» дозволяє легко змінювати розмір кроку обчислення.

4.1.2 Багатокрокові методи

В цих методах для обчислення значення нової точки використовується інформація про декілька значень, що отримані раніше. Для цього використовуються дві формули: прогнозу і корекції. Алгоритм обчислення для всіх методів прогнозу і корекції однаковий та зображений на рисунку 4.3. Вказані методи відрізняються лише формулами і не мають властивості «самостартування», оскільки вимагають знання попередніх значень. Перш ніж використовувати метод прогнозу і корекції, обчислюють початкові дані за допомогою будь-якого однокрокового методу. Часто для цього використовують метод Рунге-Кутта.

Обчислення виконують таким чином. Спочатку за формулою прогнозу та початковим значенням змінних знаходять значення $y_{n+1}^{(0)}$. Індекс (0) означає, що значення, яке прогнозується, є одним із послідовності значень y_{n+1} в міру їх уточнення. За значенням $y_{n+1}^{(0)}$ за допомогою початкового диференціального рівняння (4.1) знаходять похідну $y_{n+1}^{(0)'} = f(x_{n+1}, y_{n+1}^{(0)})$, яка після цього підставляється у формулу корекції для обчислення уточненого значення $y_{n+1}^{(j+1)}$. В свою чергу, за $y_{n+1}^{(j+1)}$ знаходять похідну $y_{n+1}^{(j+1)'} = f(x_{n+1}, y_{n+1}^{(j+1)})$. Якщо це значення не достатньо близьке до попереднього, то воно вводиться у формулу корекції і ітераційний процес продовжується. У випадку близькості значень похідних визначається y_{n+1} , яке і є остаточною. Після цього процес повторюється на наступному кроці, на якому обчислюється y_{n+2} .

Зазвичай при виведенні формул прогнозу і корекції розв'язання рівняння розглядають як процес наближеного інтегрування, а самі формули отримують за допомогою методів чисельного інтегрування.

Якщо диференціальне рівняння $y' = f(x, y)$ проінтегрувати в інтервалі значень від x_n до x_{n+k} , то результат матиме вигляд

$$y(x_{n+k}) - y(x_n) = \int_{x_n}^{x_{n+k}} f(x, y) dx.$$

Цей інтеграл не можна обчислити безпосередньо, тому що $y(x)$ – невідома функція. Вибір методу наближеного інтегрування і буде визначати метод розв'язання диференціальних рівнянь. На етапі прогнозу можна використовувати будь-яку формулу чисельного інтегрування, якщо до неї не входить попереднє значення $y'(x_{n+1})$.

В таблицю 4.1 зведені найбільш розповсюджені формули прогнозу і корекції.

Таблиця 4.1 – Найбільш розповсюджені формули прогнозу і корекції.

| Метод | Формула прогнозу | Формула корекції |
|-----------------|--|---|
| Мілна | $y_{n+1} = y_{n-3} + \frac{4}{3}h^*$ $*(2y'_n - y'_{n-1} + 2y'_{n-2}),$ $\Delta \leq \frac{28}{90}h^5 y^{(v)},$ <p>де $y' - n'$ ята похідна $f(x, y)$</p> | $y_{n+1} = y_{n-1} + \frac{1}{3}h^*$ $*(y'_{n+1} + 4y'_n + y'_{n-1}),$ $\Delta \leq \frac{1}{90}h^5 y^{(v)}$ |
| Адамса-Башфорга | $y_{n+1} = y_n + \frac{1}{24}h^*$ $*(55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}),$ $\Delta \leq \frac{251}{720}h^5 y^{(v)}$ | $y_{n+1} = y_n + \frac{1}{24}h^*$ $*(9y'_{n+1} - 19y'_n - 5y'_{n-1} + y'_{n-2}),$ $\Delta \leq \frac{19}{720}h^5 y^{(v)}$ |
| Хеммінга | $y_{n+1}^{(0)} = y_{n-3} + \frac{3}{4}h^*$ $*(2y'_n - y'_{n-1} + 2y'_{n-2})$ <p>Уточнення прогнозу</p> $\bar{y}_{n+1}^{(0)} = y_{n+1}^{(0)} +$ $+ \frac{112}{121}(y_n - y_n^{(0)})^*$ $*[\bar{y}_{n+1}^{(0)}] = f(x_{n+1}, \bar{y}_{n+1}^{(0)})$ | $y_{n+1}^{(j+1)} = \frac{1}{8}*(9y_n - y_{n-2}) +$ $+ \frac{3}{8}h^*([\bar{y}_{n+1}^{(j)}] + 2y'_n - y'_{n-1})$ |
| | $y_{n+1} = y_{n-3} + \frac{4}{3}h^*$ $*(2y'_n - y'_{n-1} + 2y'_{n-2}),$ $\Delta \leq \frac{28}{90}h^5 y^{(v)}$ | $y_{n+1} = \frac{1}{8}^*$ $*[9y_n - y_{n-2} + 3h(y'_{n+1} + 2y'_n - y'_{n-1})],$ $\Delta \leq \frac{1}{40}h^5 y^{(v)}$ |

Для більшості методів прогнозу і корекції оцінюють похибку, користуючись таким співвідношенням:

$$\Delta \leq \frac{1}{5}[y_n^{(0)} - y_n^{(j)}]$$

Мірою похибки слугує і $\epsilon y_n^{(j)}$, що входить до алгоритму рисунка 4.4.

Часто в довідниках наводяться більш точні формули для оцінювання похибки багатокрокових методів.

При виборі величини кроку можна скористатися умовою:

$$h < \frac{2}{M_2},$$

2) можна отримати оцінку похибки зрізання, не звертаючись до обчислення додаткових величин;

3) методи прогнозу і корекції не дозволяють легко змінювати крок обчислень, для цього необхідно весь ітераційний процес починати спочатку.

4.1.3 «Жорсткі» задачі

Існують звичайні диференціальні рівняння, для яких складно отримати задовільний розв'язок з використанням описаних вище методів. Виникнення таких задач зв'язано з поняттям сталої часу диференціального рівняння, проміжком часу, коли змінна частина розв'язку зменшується в e разів. Рівняння порядку n має n постійних часу; якщо будь-які дві з них суттєво (на практиці в сто і більше разів) різняться за величиною або яка-небудь з них достатньо мала порівняно з інтервалом часу, на якому відшукується розв'язок, то задача називається «жорсткою» і її практично неможливо якісно розв'язати звичайними чисельними методами. В таких випадках крок повинен бути достатньо малим, щоб можна було враховувати приріст складових розв'язку, які найбільш швидко змінюються, навіть після того, як їх внесок стане практично непомітним. Але зменшення кроку призводить до збільшення витрат часу ЕОМ і накопиченню помилок. Розроблені спеціальні методи для розв'язування таких задач, які часто зустрічаються в теорії автоматичного керування.

Найбільш простим є так званий зворотний метод Ейлера, в якому розв'язок знаходиться з рівняння, яке містить y_{n+1} у неявному вигляді:

$$y_{n+1} = y_n + hf(x_n, y_{n+1}).$$

На практиці «жорсткими» можуть виявитися рівняння, у яких коефіцієнти при похідних суттєво (в сто і більше разів) відрізняються один від одного.

4.1.4 Вибір методу розв'язання задачі Коші

Порівнюючи ефективність однокрокових і багатокрокових методів, виділяють такі особливості.

1. Багатокрокові методи вимагають більшого обсягу пам'яті ЕОМ, тому що оперують більшою кількістю початкових даних.

2. При використанні багатокрокових методів існує можливість оцінювання похибки на кроці, тому значення кроку обирається оптимальним, а в однокрокових – з деяким запасом, що знижує швидкодію.

3. При однаковій точності багатокрокові методи вимагають меншого обсягу обчислень. Наприклад, в методі Рунге-Кутта четвертого порядку точності доводиться обчислювати чотири значення функції на кожному кроці, а для забезпечення збіжності методу прогнозу і корекції того ж порядку точності – достатньо двох.

4. Однокрокові методи на відміну від багатокрокових дозволяють одразу почати розв'язання задачі («самостартування») і легко змінювати крок в процесі обчислень.

Перед початком розв'язання задачі необхідно провести перевірку на «жорсткість» і у випадку позитивного результату використати спеціальні методи. Якщо задача Коші дуже складна, то зазвичай перевага надається методу прогнозу і корекції, який має до того ж більш високу швидкодію. Початок розв'язання задачі при цьому проводиться за допомогою однокрокових методів. Якщо для обчислення чергового значення y_i вимагається більш ніж дві ітерації або якщо помилка зрізання дуже велика, то необхідно зменшити крок h . З іншого боку, при дуже малій похибці зрізання можна збільшити крок, тим самим підвищити швидкодію, але при цьому весь процес розв'язання треба починати спочатку. Інколи на практиці вимагається мінімізувати час підготовки задачі до розв'язання. Тоді доцільно використовувати методи Рунге-Кутта.

На закінчення слід відзначити, що велике значення для ефективного розв'язання задачі мають досвід, інтуїція і кваліфікація користувача як при постановці задачі, так і в процесі вибору методу розробки алгоритму і програми для ЕОМ. При цьому часто зручно користуватись вже готовими програмними засобами, які є в наявності (наприклад, в пакетах Maple, Mathematika).

4.2 Методи розв'язання крайових задач

Методи розв'язання крайових задач розглядаються на прикладі звичайного диференціального рівняння другого порядку

$$\frac{d^2y}{dx^2} = f(x, y, \frac{dy}{dx})$$

при граничних умовах $y(a) = A$ $y(b) = B$. Методи розв'язання

крайових задач розділяють на дві групи: методи, що побудовані на заміні розв'язання крайової задачі розв'язанням декількох задач Коші (методи «стрілянини») та різницеві методи.

4.2.1 Метод «стрілянини»

Якщо звичайне диференціальне рівняння другого порядку – лінійне, то воно має вигляд:

$$y'' = f_1(x)y' + f_2(x)y + f_3(x)$$

при $y(a) = A$ $y(b) = B$.

Крайову задачу можна звести до задачі Коші введенням додаткової початкової умови, крім $y(a)=A$ вводиться $y'(a)=\alpha_1$.

Знайшовши розв'язок $y_1(x)$, можна поставити іншу початкову умову $y'(a)=\alpha_2$ і отримати інший розв'язок $y_2(x)$. Якщо $y_1(b)=\beta_1$ а $y_2(b)=\beta_2$, причому $\beta_1 \neq \beta_2$, то розв'язок:

$$y(x) = \frac{1}{\beta_1 - \beta_2} [(B - \beta_2)y_1(x) + (\beta_1 - B)y_2(x)]$$

буде задовольняти обидві початкові умови.

При розв'язуванні нелінійного звичайного диференціального рівняння методами «стрілянини» крайова задача зводиться до розв'язування декількох задач Коші, послідовно вводячи в початкові умови значення α :

$$y(a)=A \text{ і } y'(a)=\alpha$$

і намагаючись знайти розв'язок, який задовольняє умову $y(b)=B$.

При цьому алгоритм досягнення мети будується на основі одного з методів оптимізації. Однак цей шлях розв'язання задачі пов'язаний з великими обчислювальними труднощами, і тому у випадку нелінійних диференціальних рівнянь перевага надається різницеvim методам.

4.2.2 Різницеві методи

Апарат різницевих методів (difference methods) являє собою потужний засіб чисельного розв'язування звичайних диференціальних рівнянь і диференціальних рівнянь у частинних похідних. У його основі лежить подання незалежного аргументу на відрізку $[a, b]$ у вигляді дискретної множини точок x_i , $i=0, \dots, n$, $x_0=a$, $x_n=b$, яка називається сіткою.

Найбільше поширення отримала рівномірна сітка з кроком $x_i - x_{i-1} = h$. При цьому замість безперервної функції $f(x)$ розглядається сіткова функція $y_i = f(x_i)$. Аналогічно проводиться дискретизація функції багатьох змінних, наприклад, двох:

$$x_{ij}, i=0, \dots, n, j=0, \dots, m, y_{ij}=f(x_{ij}).$$

Крім найбільш розповсюдженої прямокутної сітки використовують полярну, трикутну, скошену та інші, зображені на рисунку 4.5. Багатовимірні сітки знаходять використання в задачах з частинними похідними за декількома незалежними змінними.

Розв'язання задачі різницевиими методами складається з двох етапів:

- отримання дискретної (різницевої) апроксимації диференціальних рівнянь і дослідження отриманих при цьому різницевих рівнянь;
- розв'язання різницевих рівнянь.

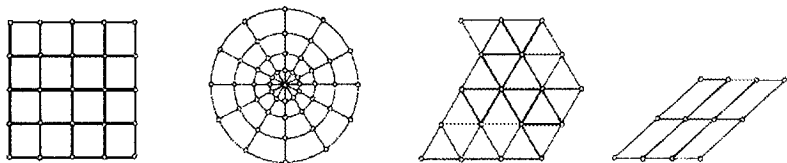


Рисунок 4.5 – Двовимірні сітки

При отриманні різницевих схем важливу роль відіграє загальна вимога: щоб різницева схема якомога краще наближала основні властивості початкового диференціального рівняння. Такі різницеві схеми можна отримати за допомогою варіаційних принципів та інтегральних співвідношень. Оцінювання точності різницевої схеми зводиться до вивчення похибки апроксимації та стійкості. Сіткову функцію можна розглядати як функцію цілочислового аргументу

$$y(i)=y_i, \quad i=0, \pm 1, \pm 2, \dots$$

Для y_i можна ввести операції, які є дискретним (різницевим) аналогом операцій диференціювання та інтегрування.

Аналогом першої похідної є різниця першого порядку:

$$\Delta y_i = y_{i+1} - y_i - \text{права різниця};$$

$$\nabla y_i = y_i - y_{i-1} - \text{ліва різниця};$$

$$\delta y_i = \frac{1}{2}(\Delta y_i + \nabla y_i) = \frac{1}{2}(y_{i+1} - y_{i-1}) - \text{центральна різниця}.$$

$$\text{Зауважимо, що } \Delta y_i = \nabla y_{i+1}.$$

Далі можна записати різниці другого порядку

$$\Delta^2 y_i = \Delta(\Delta y_i) = \Delta(y_{i+1} - y_i) = y_{i+2} - 2y_{i+1} + y_i,$$

$$\Delta \nabla y_i = \Delta(y_i - y_{i-1}) = (y_{i+1} - y_i) - (y_i - y_{i-1}) = y_{i+1} - 2y_i + y_{i-1},$$

$$\text{тобто } \Delta^2 y_i = \Delta \nabla y_{i+1}.$$

Аналогічно визначається різниця m -го порядку:

$$\Delta^m y_i = \Delta(\Delta^{m-1} y_i).$$

Звідси виходить, що

$$\sum_{j=k}^i \Delta y_j = y_{i+1} - y_k, \quad \sum_{j=k}^i \nabla y_j = y_i - y_{k-1}.$$

На множині вузлів сітки, яка називається шаблоном (тут розглядаються одновимірні шаблони, в наступному розділі вводяться у розгляд двовимірні), замінимо неперервний диференціальний оператор Ly різницеvim оператором $L_h y$.

Наприклад, різницеві оператори для першої похідної на трьох вузлах сітки $(x-h, x, x+h)$:

$$L_h^+ y = \frac{y(x+h) - y(x)}{h} = y_x^+$$

$$L_h^- y = \frac{y(x) - y(x-h)}{h} = y_x^-$$

$$L_h^0 y = \frac{y(x+h) - y(x-h)}{2h} = y_x^0$$

права, ліва і центральна, відповідно, різницеві похідні.

Аналогічно для другої похідної

$$\begin{aligned} L_h y &= \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} = \frac{y_x^+(x) - y_x^-(x)}{h} = \\ &= \frac{y_x^-(x+h) - y_x^-(x)}{h} = y_{xx}^-(x). \end{aligned}$$

При розв'язуванні крайової задачі записуються різницеві рівняння для всіх n вузлів області змінення $x \in [a, b]$. Враховуючи дві граничні умови $y_0 = y(a)$ та $y_n = y(b)$, отримують систему з $n-1$ алгебраїчних рівнянь з $n-1$ невідомими y_i . Якщо початкове звичайне диференціальне рівняння лінійне, то задача зводиться до розв'язування системи лінійних алгебраїчних рівнянь, а якщо нелінійне – то нелінійних або трансцендентних алгебраїчних систем. Привести крайову задачу, яку можна розв'язати методом кінцевих різниць, до вигляду, зручного для розробки стандартної програми для ЕОМ, складно, оскільки формулювання кожної задачі залежить від вигляду диференціального рівняння, яке розглядають.

В цьому розділі поданий лише короткий опис різницевого підходу без деталізації методів побудови і дослідження різницевих схем. Взагалі різницеві методи є універсальними методами чисельного аналізу, хоча й

розглядаються в більшості випадків у зв'язку з розв'язанням крайових задач і диференціальних рівнянь в частинних похідних. Наприклад, метод Ейлера розв'язання задачі Коші може бути інтерпретований як використання одновимірного різницевого оператора. Прикладом узагальненого підходу до розв'язання різноманітних задач обчислювальної математики різницевиими методами можуть слугувати праці академіка А. А. Самарського, що містять детальний аналіз і дослідження питань теорії і практики використання різницевих схем.

Контрольні запитання та завдання

1. Сформулюйте задачу Коші і крайову задачу. В чому відмінність у постановці цих задач?
2. Наведіть приклади звичайних диференціальних рівнянь, які розв'язуються лише чисельними методами.
3. Дайте порівняльну оцінку і визначте області застосування методів розв'язання звичайних диференціальних рівнянь.
4. Чи можлива крайова задача для рівнянь першого порядку?
5. Що таке похибка зрізання (обмеження)? Чим визначається порядок похибки методу?
6. Покажіть, що загальна формула для методу Рунге-Кутта, застосована до рівняння $y' = -y$ з початковою умовою $y(0) = 1$, приводить до співвідношення $y_n = (1 - h + \frac{h^2}{2})^n \approx e^{-x_n}$.
7. В чому подібність між методами прогнозу і корекції і виправленим методом Ейлера?
8. Як оцінюються похибки методів Ейлера, Рунге-Кутта, прогнозу і корекції?
9. Дайте геометричну інтерпретацію методів Ейлера.
10. Що таке властивість «самостартування» методів розв'язання звичайних диференціальних рівнянь?
11. Як обирається крок при використанні багатокрокових методів розв'язання?
12. Які рівняння називають «жорсткими»? У чому особливості їх розв'язання?
13. Для розв'язання яких крайових задач може використовуватись метод «стрілянини»? Як отримати розв'язок крайової задачі з розв'язків декількох задач Коші в цьому методі?
14. З яких етапів складається розв'язання задачі різницевиими методами?
15. Запишіть всі можливі різницеві оператори для першої і другої похідних.
16. Як визначаються різницеві оператори для похідних більш високих порядків?

РОЗДІЛ 5 МЕТОДИ РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ В ЧАСТИННИХ ПОХІДНИХ

З розв'язанням диференціальних рівнянь в частинних похідних (differential equations in partial derivatives) інженерам і дослідникам доводиться зустрічатися у багатьох галузях науки і техніки, в аеро- і гідродинаміці, ядерній фізиці, радіозв'язку. Математичні моделі з диференціальними рівняннями в частинних похідних широко використовуються в теорії автоматичного керування і у вимірювальній техніці. В таких рівняннях містяться частинні похідні і шукана величина залежить відразу від декількох змінних. Розглянемо диференціальне рівняння другого порядку з двома незалежними змінними:

$$\begin{aligned} A(x, y) \frac{\partial^2 f}{\partial x^2} + B(x, y) \frac{\partial^2 f}{\partial x \partial y} + C(x, y) \frac{\partial^2 f}{\partial y^2} + \\ E(x, y) \frac{\partial f}{\partial y} + D(x, y) \frac{\partial f}{\partial x} + F(x, y) = G(x, y). \end{aligned} \quad (5.1)$$

Аналогічно звичайним диференціальним рівнянням єдиний розв'язок рівняння (5.1) можна отримати лише задавши додаткові умови, але оскільки тут присутні дві незалежні змінні x та y , умова повинна задаватися для якої-небудь кривої у площині xy . Ця умова може бути накладена на функцію f або (та) на її похідні та залежати від типу рівняння, яке визначає її вигляд і характер зміни.

Існують три типи диференціальних рівнянь другого порядку:

- еліптичні, при $B^2 - 4AC < 0$;
- параболічні, при $B^2 - 4AC = 0$;
- гіперболічні, при $B^2 - 4AC > 0$.

Рівняння можуть переходити з одного типу в інший в залежності від значень коефіцієнтів.

Еліптичні рівняння описують усталені (стаціонарні) процеси, причому задача ставиться в замкненій області, і в кожній точці границі цієї області задаються граничні умови. Інші два типи рівнянь описують еволюційні процеси. В таких задачах найбільш поширений випадок, коли на одній частині границі ставлять граничні умови, а на іншій – початкові.

Приклади деяких диференціальних рівнянь в частинних похідних, які описують різні типи задач, наведені в таблиці 5.1.

Таблиця 5.1 – Диференціальні рівняння в частинних похідних

| Рівняння | Математична форма | Приклади задач, в яких зустрічається рівняння |
|--------------|--|--|
| Лапласа | $\Delta f = 0$ | Усталена течія рідини. Стационарні теплові поля |
| Пуассона | $\Delta f = -R$ | Теплопередача з внутрішнім джерелом тепла |
| Дифузії | $\Delta f = \frac{1}{h^2} \frac{\partial^2 f}{\partial t^2}$ | Нестационарна теплопровідність |
| Хвильове | $\Delta f = \frac{1}{c^2} \frac{\partial^2 f}{\partial t^2}$ | Розповсюдження хвиль |
| Бігармонічне | $\Delta^2 f = F(x, y)$ | Деформація пластин |

В таблиці використані прийняті позначення найбільш поширених операторів:

$$\text{оператор Лапласа } \Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2};$$

$$\text{бігармонічний оператор } \Delta^2 f = \frac{\partial^4 f}{\partial x^4} + 2 \frac{\partial^4 f}{\partial x^2 \partial y^2} + \frac{\partial^4 f}{\partial y^4}.$$

Існують два методи розв'язання диференціальних рівнянь в частинних похідних: різницевий метод (метод скінченних різниць) і метод скінченних елементів. В сучасній прикладній математиці обидва методи розглядаються як інтерпретації використання загальної теорії різницевого схем до розв'язання диференціальних рівнянь в частинних похідних.

В основі методу скінченних елементів лежить варіаційне обчислення. Диференціальне рівняння, яке описує задачу й відповідні граничні умови, використовується для постановки варіаційної задачі. В методі скінченних елементів фізична задача замінюється кусково-гладкою моделлю. Цей метод вимагає складної постановки задачі, високої кваліфікації й досвіду, неуніверсальний (кожне розв'язання застосовується лише для конкретної задачі). Метод скінченних елементів знайшов широке використання для розв'язання спеціальних задач в теоретичній механіці, гідродинаміці, теорії поля, він складний, вимагає серйозної підготовки і знань в конкретній галузі використання, для його достатньо повного викладу довелося б написати спеціальний підручник. Тому в даній роботі не наводиться докладний виклад методу скінченних елементів, тим більше, що при розв'язанні задач автоматизації та систем керування частіше використовується різницевий метод.

5.1 Різницьвий метод

Для диференціальних рівнянь другого порядку в частинних похідних найчастіше використовується двовимірна прямокутна сітка. Центральнорізницьві шаблони, що їх застосовують на двовимірній квадратній сітці з кроком h , зображеній на рисунку 5.1 (індекс j надається незалежній змінній y , а i стосується x), можуть бути отримані аналогічно одновимірному випадку.

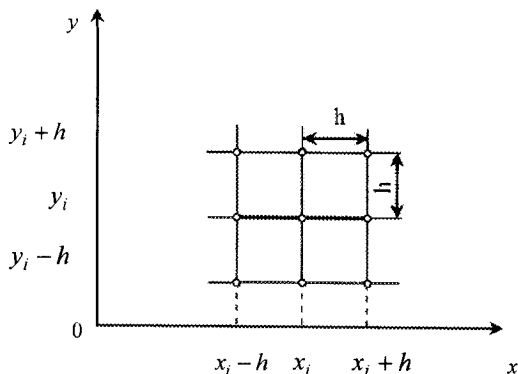


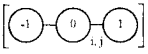
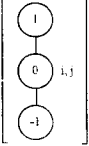
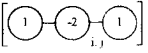
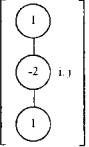
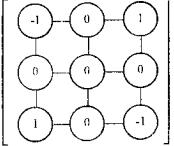
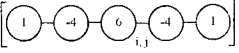
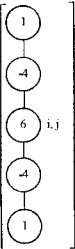
Рисунок 5.1 – Квадратна сітка

Для зручності позначення $f(x_i + h, y_i)$ замінимо на $f_{i+1,j}$. Користуючись цим позначенням, отримаємо вирази для частинних похідних, з якими доводиться зустрічатися на практиці й використання яких ілюструється відповідними обчислювальними шаблонами. Обчислювальні шаблони для похідних наведені в таблиці 5.2.

З цих елементів будуються більш складні обчислювальні шаблони для диференціальних рівнянь. Додавання похідних здійснюється суперпозицією відповідних обчислювальних шаблонів. Цим методом конструюються шаблони для Δf і $\Delta^2 f$ (рисунок 5.2).

Всі наведені обчислювальні шаблони мають похибку другого порядку. Можна побудувати більш точні обчислювальні шаблони, якщо внести у розгляд додаткові вузли. Іноді, щоб звести до мінімуму розповсюдження похибок, користуються лівими або правими різницями.

Таблиця 5.2 – Обчислювальні шаблони для похідних

| | |
|--|---|
| $\frac{\partial f}{\partial x} \approx \frac{f_{i+1,j} - f_{i-1,j}}{2h} \approx \frac{1}{2h}$ |  |
| $\frac{\partial f}{\partial y} \approx \frac{f_{i,j+1} - f_{i,j-1}}{2h} \approx \frac{1}{2h}$ |  |
| $\frac{\partial^2 f}{\partial x^2} \approx \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h^2} \approx \frac{1}{h^2}$ |  |
| $\frac{\partial^2 f}{\partial y^2} \approx \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h^2} \approx \frac{1}{h^2}$ |  |
| $\frac{\partial^2 f}{\partial x \partial y} \approx \frac{f_{i+1,j+1} - f_{i-1,j+1} - f_{i+1,j-1} + f_{i-1,j-1}}{4h^2} \approx \frac{1}{4h^2}$ |  |
| $\frac{\partial^4 f}{\partial x^4} \approx \frac{1}{h^4}$ |  |
| $\frac{\partial^4 f}{\partial y^4} \approx \frac{1}{h^4}$ |  |

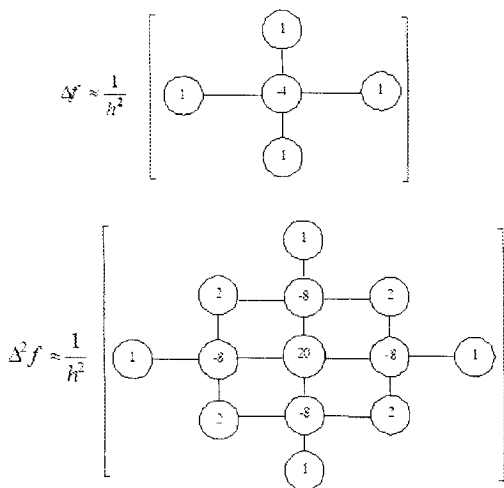


Рисунок 5.2 – Обчислювальні шаблони для операторів

Часто труднощі, що пов'язані з використанням прямокутної сітки, виникають через границю неправильної конфігурації, яка не проходить через вузли сітки. Розглянемо приклад розв'язання такої задачі для обчислювального шаблону рівняння Лапласа в області, що обмежена довільною кривою, яка зображена на рисунку 5.3.

Другі частинні похідні для вузлів, які лежать на границі області, можна записати у вигляді:

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{\frac{f_a - f_{i,j}}{ah} - \frac{f_{i,j} - f_{i-1,j}}{h}}{0,5(ah + h)}$$

$$\frac{\partial^2 f}{\partial y^2} \approx \frac{\frac{f_b - f_{i,j}}{bh} - \frac{f_{i,j} - f_{i,j-1}}{h}}{0,5(bh + h)}$$

Додавши другі похідні, отримаємо :

$$\Delta f \approx \frac{2}{h^2} \left(\frac{f_{i-1,j}}{1+a} + \frac{f_a}{a(1+a)} + \frac{f_b}{b(1+b)} + \frac{f_{i,j-1}}{1+b} - \frac{a+b}{ab} f_{i,j} \right).$$

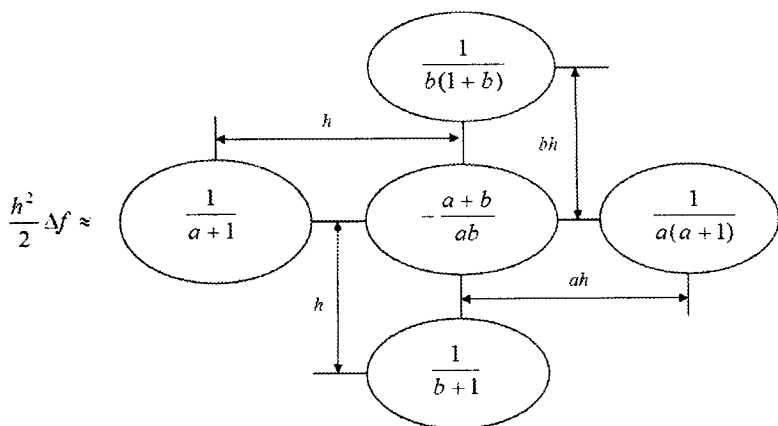
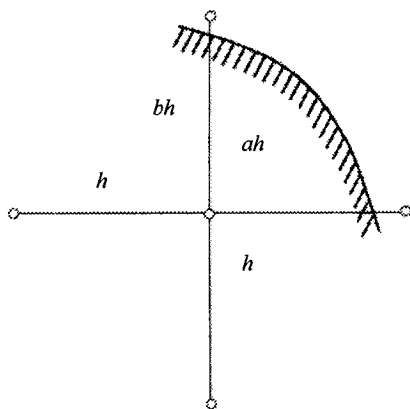


Рисунок 5.3 – Обчислювальний шаблон для границі неправильної форми

Застосувавши обчислювальний шаблон до кожного з n вузлів сітки, отримаємо систему з n рівнянь, яка може бути лінійною, якщо початкове диференціальне рівняння має відповідну структуру. В цьому випадку розв'язання задачі зводиться до розв'язання системи рівнянь вигляду:

$$\begin{bmatrix} \text{матриця} \\ \text{коефіцієнтів} \end{bmatrix} \begin{bmatrix} \text{невідоме значення } u \\ \text{вузлах (вектор - стовпець)} \end{bmatrix} = \begin{bmatrix} \text{вектор - стовпець} \\ \text{вільних членів} \end{bmatrix},$$

яка розв'язується найчастіше ітераційними методами.

5.2 Розв'язання різних типів диференціальних рівнянь в частинних похідних

Практичні методи і алгоритми розв'язання різних типів диференціальних рівнянь в частинних похідних мають свої особливості і вимагають окремого розгляду на прикладі найбільш розповсюджених задач.

5.2.1 Еліптичні рівняння

До еліптичних рівнянь (elliptic equations) приводиться багато різних фізичних задач: розрахунок напружень, які виникають при пружному скручуванні довгого циліндричного стрижня; розподіл електричних напруг на площині, що проводить струм; задача про стаціонарні течії тепла в двовимірному тілі. Часто виникає необхідність розв'язання таких задач і в теорії автоматичного керування. Більшість еліптичних рівнянь описується рівнянням Пуассона або його окремим випадком – рівнянням Лапласа.

Розглянемо класичну задачу Діріхле для рівняння Лапласа в прямокутній області, яка формулюється таким чином: знайти неперервну функцію $f(x,y)$, яка задовольняє $\Omega = \{(x,y) / 0 \leq x \leq a, 0 \leq y \leq b\}$ всередині прямокутної області рівняння Лапласа:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$$

і приймає на границі області задані значення:

$$\begin{aligned}x=0; & f(0,y) = f_1(y), \\x=a; & f(a,y) = f_2(y), \\y=0; & f(x,0) = f_3(x), \\y=b; & f(x,b) = f_4(x).\end{aligned}$$

Введемо в області розв'язання двовимірну сітку з кроком h по осі x і l по осі y . Тоді, користуючись прийнятими в попередніх розділах позначеннями і апроксимуючи рівняння Лапласа різницеvim рівнянням, отримаємо таку систему лінійних рівнянь (приймемо для спрощення $l=h$):

$$\begin{aligned}f_{ij} &= \frac{1}{4}(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}), \\f_{i,0} &= f_3(x_i), \quad f_{i,m} = f_4(x_i), \quad f_{0,j} = f_1(y_j), \quad f_{n,j} = f_2(y_j)\end{aligned}\tag{5.2}$$

при $i=1,2,\dots,n-1; j=1,\dots,m-1$.

Ця система рівнянь має велику кількість нульових елементів і задовольняє умову збіжності при використанні ітераційних методів. Найбільше використання для розв'язання таких систем знайшов метод Гаусса-Зейделя, який, коли застосовується до еліптичних різницевих рівнянь, називається методом Лібмана або методом послідовних зміщень. Порядок ітерацій можна простежити, переписавши систему (5.2.) у вигляді:

$$f_{1,1}^{(m+1)} = \frac{1}{4} [f_3(h) + f_1(h) + f_{2,1}^{(m)} + f_{1,2}^{(m)}],$$

$$f_{2,1}^{(m+1)} = \frac{1}{4} [f_3(2h) + f_{1,1}^{(m+1)} + f_{3,1}^{(m)} + f_{2,2}^{(m)}],$$

$$f_{3,1}^{(m+1)} = \frac{1}{4} [f_3(3h) + f_{2,1}^{(m+1)} + f_{4,1}^{(m)} + f_{3,2}^{(m)}],$$

$$f_{n-1,1}^{(m+1)} = \frac{1}{4} \{f_3[(n-1)h] + f_2(h) + f_{n-2,1}^{(m+1)} + f_{n-1,2}^{(m)}\},$$

$$f_{1,2}^{(m+1)} = \frac{1}{4} [f_1(2h) + f_{1,1}^{(m+1)} + f_{2,2}^{(m)} + f_{1,3}^{(m)}],$$

$$f_{2,2}^{(m+1)} = \frac{1}{4} [f_{2,1}^{(m+1)} + f_{1,2}^{(m+1)} + f_{3,2}^{(m)} + f_{2,3}^{(m)}],$$

де верхніми індексами позначено порядковий номер ітерації: m – попередня, $m+1$ – наступна.

Зазвичай вважають $f_{i,j}^{(0)} = 0$ для всіх i, j . Система рівнянь легко розв'язується на ЕОМ. Взагалі кажучи, будь-які еліптичні рівняння, які не

містять $\frac{\partial^2 f}{\partial x \partial y}$, зводяться до систем різницевих рівнянь, які можна

розв'язувати як методом Лібмана, так і іншими ітераційними методами (Якобі, послідовної верхньої релаксації та ін.), оскільки для них

виконуються умови збіжності. Для еліптичних рівнянь, які містять $\frac{\partial^2 f}{\partial x \partial y}$ в

загальному вигляді, питання про збіжність ітераційних методів не має теоретичного розв'язку і необхідно розглядати отриману систему рівнянь в кожному конкретному випадку.

5.2.2 Гіперболічні рівняння

Найчастіше зустрічається в інженерній практиці гіперболічне рівняння в частинних похідних (hyperbolic equations in partial derivatives) – хвильове рівняння, яке описує різні види коливань: коливання струни або мембрани, розповсюдження звукових хвиль в різних середовищах тощо.

В загальному вигляді задача формулюється таким чином: знайти функцію $f(x, t)$, яка задовольняє всередині області $\Omega = \{(x, t), 0 \leq x \leq a, 0 \leq t \leq T\}$ рівняння

$$\frac{\partial^2 f}{\partial t^2} = c^2 \frac{\partial^2 f}{\partial x^2} \quad (c = \text{const} > 0),$$

початкові

$$f(x, 0) = f_0(x),$$

$$\left. \frac{\partial f}{\partial t} \right|_{x,0} = g(x)$$

і граничні умови

$$f(0, t) = \mu_1(t),$$

$$f(a, t) = \mu_2(t).$$

Оскільки заміна змінних $t = c \tau$ приводить рівняння до вигляду:

$$\frac{\partial^2 f}{\partial \tau^2} = \frac{\partial^2 f}{\partial x^2},$$

то надалі приймаємо $c=1$.

Переходячи до різницевого рівняння на сітці з кроком h по x й τ по t з центральними різницями, отримасмо

$$\frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h^2} = \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\tau^2}.$$

Якщо ввести $r = \frac{\tau}{h}$, то вираз для $f_{i,j+1}$ набуде вигляду:

$$f_{i,j+1} = r^2 (f_{i+1,j} + f_{i-1,j}) + 2(1 - r^2) f_{i,j} - f_{i,j-1}. \quad (5.3)$$

Схема розв'язання за рівнянням (5.3) називається тришаровою, оскільки зв'язує значення $f_{i,j}$ на трьох часових шарах $j-1, j, j+1$. Ця схема явна й дозволяє в явному вигляді виразити $f_{i,j}$ через значення f з попередніх шарів (існують неявні схеми, основані на використанні інших обчислювальних шаблонів, але вони вимагають більшого обсягу обчислень при розв'язанні системи рівнянь). Для знаходження розв'язку на першому шарі зазвичай використовують інтерполяційні методи. Наприклад,

$$f_{i,1} = f_{i,0} + \tau g(x_i). \quad (5.4)$$

Співвідношення сторін сітки визначається величиною r , яка є мірою стійкості отриманого розв'язку. При $r > 1$ розв'язок нестійкий, при $r < 1$ хоча й стійкий, але точність його при зменшенні r спадає, при $r = 1$ різницевий розв'язок стійкий і збігається з точним. Вибір $r = 1$ зручний ще й тим, що при цьому спрощується співвідношення (5.4)

$$f_{i,j+1} = f_{i+1,j} + f_{i-1,j} - f_{i,j-1}.$$

5.2.3 Параболічні рівняння

Прикладом задачі, яка приводить до параболічного рівняння (parabolic equation) в частинних похідних, є задача про теплопередачу по довгому стрижню. Вона описується рівнянням теплопередачі (або дифузії).

Задача полягає у знаходженні $f(x, t)$, яка задовольняє в області $\Omega = \{(x, t) \mid 0 \leq x \leq a, 0 \leq t \leq T\}$ рівняння

$$\frac{\partial f}{\partial t} = k \frac{\partial^2 f}{\partial x^2} \quad (k = \text{const} > 0),$$

початкові

$$f(x, 0) = f_0(x)$$

і граничні умови першого роду

$$f(0, t) = \mu_1(t),$$

$$f(a, t) = \mu_2(t).$$

Заміна змінних $t = k t$ приводить рівняння до вигляду

$$\frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial x^2},$$

тому надалі будемо вважати $k=1$.

Можливі два варіанти отримання різницевого рівняння на сітці з кроком h по x та τ по t (рисунок 5.4).

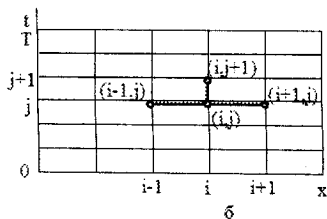
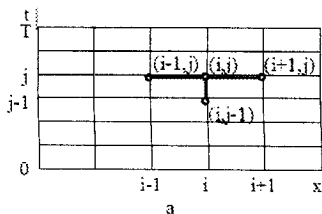


Рисунок 5.4 – Обчислювальні шаблони для параболічних рівнянь

Варіант з апроксимацією на чотириточковому шаблоні (рисунок 5.4, а) приводить до неявної двошарової різницевої схеми

$$2f_{i+1,j} - (1 + 2r)f_{i,j} + 2f_{i-1,j} = -f_{i,j-1},$$

$$\text{де } r = \frac{\tau}{h^2}.$$

Ця схема, доповнена рівняннями, отриманими з крайових умов

$$f_{0,j} = \mu_1(t_j); \quad f_{n,j} = \mu_2(t_j),$$

зводить задачу до розв'язання системи рівнянь, які мають стійкий розв'язок при будь-яких значеннях r .

Варіант з апроксимацією на чотириточковому шаблоні (рисунок 5.4, б) приводить до явної двошарової системи

$$f_{i,j+1} = rf_{i+1,j} + (1 - 2r)f_{i,j} + 2f_{i-1,j}.$$

Ця схема стійка тільки при $r \leq 0,5$, що приводить до необхідності проводити обчислення з дуже малим кроком по t , який обмежує швидкість і вимагає більших витрат часу ЕОМ. Тому для параболічних рівнянь більш широке розповсюдження отримала неявна схема.

5.3 Загальні рекомендації до розв'язання диференціальних рівнянь в частинних похідних

На початковому етапі вибирається метод розв'язання задачі. Зазвичай простіше використовувати різницевий метод, який вимагає більш простої підготовки задачі до розв'язання, але в ряді випадків для задач з добре

розробленою теорією (наприклад, задач механіки) доцільно звертатися до методу скінченних елементів.

При визначенні кроку розв'язання задачі основним фактором є точність (якщо висока, то необхідна або дуже дрібна сітка, або розбиття на дуже малі елементи). При цьому необхідно враховувати, що похибка різницьових методів має другий порядок. Її оцінювання можна проводити аналогічно звичайним диференціальним рівнянням за методом Рунге.

У випадку симетрії в області розв'язання можна число вузлів зменшити в два або навіть у чотири рази (при симетрії по обох осях координат). Це дозволяє отримати економію часу та обсягу обчислень.

Велику роль для ефективного розв'язання задачі грає вибір початкових наближень. При використанні ітераційних методів від цього значною мірою залежить швидкість збігання. Часто має сенс розв'язувати задачу в декілька етапів: на першому за допомогою грубої сітки (або розбиття на крупні елементи) отримують добре початкове наближення, на наступних – шукають більш точний розв'язок на дрібній сітці.

Контрольні запитання та завдання

1. Наведіть приклади інженерних задач, які описуються диференціальними рівняннями в частинних похідних.

2. На які види можна поділити диференціальні рівняння другого порядку в частинних похідних в залежності від коефіцієнтів, які входять до них?

3. Які методи існують для розв'язання диференціальних рівнянь в частинних похідних?

4. Як конструюються обчислювальні шаблони для частинних похідних? Отримайте обчислювальні шаблони для оператора Лапласа та бігармонічного оператора.

5. Як використовується різницьовий метод у випадку, коли границя області, де шукають розв'язок, не проходить через вузли сітки? Наведіть приклад обчислювального шаблону?

6. Якими різницьовими методами розв'язується класична задача Діріхле для рівнянь Лапласа в прямокутній області? Доведіть їх збіжність.

7. Отримайте тришарову схему розв'язання гіперболічних рівнянь. В яких випадках розв'язок стійкий?

8. Порівняйте ефективність використання явної і неявної схем розв'язання параболічних рівнянь в частинних похідних.

9. Як вибирається крок при розв'язанні диференціальних рівнянь в частинних похідних?

РОЗДІЛ 6 МЕТОДИ ОБРОБКИ ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ

В цьому розділі розглянуті найбільш важливі задачі обробки даних (інтерполяція, апроксимація, статистична обробка), які зустрічаються в задачах проектування комп'ютеризованих систем управління.

6.1 Інтерполяція

Мета інтерполяції (interpolation) – побудувати функцію $F(x)$, яка приймає в окремих точках $x_i [a, b]$ ($i = 0, 1, 2, \dots, n$) (вузли інтерполяції) значення

$$F(x_0) = y_0, F(x_1) = y_1, \dots, F(x_i) = y_i, \dots, F(x_n) = y_n, \quad (6.1)$$

що збігаються з раніше заданими значеннями в цих точках невідомої функції $y = f(x)$. Геометрично це означає, що потрібно знайти криву $y = F(x)$ певного типу, яка проходить через систему точок $M(x_i, y_i)$ ($i = 0, 1, 2, \dots, n$) (рис. 6.1).

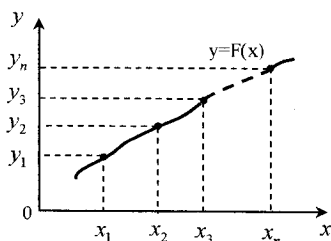


Рисунок 6.1 – Інтерполяція

В загальних випадках ця задача має нескінченну множину розв'язків чи зовсім не має розв'язку, але вона стає однозначною, якщо замість довільної функції $F(x)$ шукати поліном $P_n(x)$ ступеня не вище n , який задовольняє умову (6.1), тобто

$$P_n(x_0) = y_0, P_n(x_1) = y_1, \dots, P_n(x_i) = y_i, \dots, P_n(x_n) = y_n.$$

Інтерполяційну формулу $y = F(x)$, як правило, використовують для наближеного обчислення значень даної функції $f(x)$ для $x \neq x_i$ ($i = 0, 1, 2, \dots, n$). Така операція називається інтерполяцією. Треба

відзначити, що є інтерполяція в вузькому розумінні, коли $x \in [x_0, x_n]$, та екстраполяція, коли x знаходиться за межами інтервалу $[x_0, x_n]$ тобто $x < x_0$ чи $x > x_n$.

Говорячи про процедуру інтерполювання, обов'язково потрібно вказати обмеження, які накладаються на набір базових точок. Початкова сітка точок повинна описувати лише плавну (гладеньку) функцію. Відповідно до умов конкретної задачі обов'язково повинні задаватися значення похідної функції у крайових точках вхідної сітки для отримання однозначного результату.

До основних методів інтерполяції належать:

1) *лінійна інтерполяція (linear interpolation)*. Найбільш простий і швидкий метод, в якому задані вузлові точки з'єднуються прямими;

2) *інтерполяція із застосуванням багаточлена (interpolation using polynomial)*. Використовується багаточлен n -го порядку, який в загальному випадку має вигляд

$$P(x) = P_n(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$$

де $a_i, i = \overline{0, n}$ – постійні коефіцієнти. Всі методи знаходження інтерполяційного багаточлена зводяться до отримання постійних коефіцієнтів. До таких методів належать:

- a. Інтерполяція різницеvими методами;
- b. Ермітова поліноміальна інтерполяція;
- c. Інтерполяція Лагранжа;

3) *поліноміальна інтерполяція сплайнами (polynomial spline interpolation)*. Вузлові точки з'єднуються багаточленами заданого порядку, який обирається в залежності від методу. До найбільш поширених методів інтерполяції сплайнами належать:

- a. Класичні кубічні сплайни;
- b. Ермітові сплайни;
- c. В-сплайни;
- d. Криві Без'є.

Розглянемо декілька методів інтерполяції.

6.1.1 Різницеvі методи

Існує багато кінцево-різницеvих методів інтерполяції. Найбільш поширеним є метод Ньютона для інтерполяції «уперед» (метод Ньютона-Грегорі). Інтерполяційний поліном в цьому випадку має вигляд:

$$P_n(x) = C_0 + C_1(x - x_0) + C_2(x - x_0)(x - x_1) + \dots$$

$$\dots + C_n(x-x_0)(x-x_1) \dots (x-x_{n-1}).$$

Коефіцієнти C_i знаходять з рівнянь:

$$P_n(x_i) = y_i, \quad i = 0, 1, 2, \dots, n,$$

які дозволяють записати систему

$$\begin{cases} C_0 = y_0, \\ C_0 + C_1(x_1 - x_0) = y_1, \\ C_0 + C_1(x_2 - x_0) + C_2(x_2 - x_0)(x_2 - x_1) = y_2, \\ \dots \\ C_0 + C_1(x_n - x_0) + \dots + C_n(x_n - x_0)(x_n - x_1) \dots (x_n - x_{n-1}) = y_n. \end{cases} \quad (6.2)$$

Це лінійна система рівнянь з трикутною матрицею.

Якщо прийняти крок $x_{i+1} - x_i = h$, то в області змінювання $x \in [x_0, x_n]$ отримаємо одновимірну рівномірну сітку та зможемо скористатись різницеvim зображенням системи (6.2), яке приводить до таких різницеvих виразів для коефіцієнтів:

$$C_0 = y_0,$$

$$C_1 = \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h},$$

Δy_0 – права різниця першого порядку в точці y_0 ;

$$C_2 = \frac{y_2 - 2y_1 + y_0}{2h^2} = \frac{\Delta^2 y_0}{2h^2},$$

$\Delta^2 y_0$ – права різниця другого порядку;

$$C_j = \frac{\Delta^j y_0}{(j!)h^j},$$

$\Delta^j y_0$ – права різниця j -го порядку.

Тоді

$$\begin{aligned} P_n(x) = y_0 + \frac{\Delta y_0}{1!h}(x-x_0) + \frac{\Delta^2 y_0}{2!h^2}(x-x_0)(x-x_1) + \dots \\ \dots + \frac{\Delta^n y_0}{n!h^n}(x-x_0)(x-x_1) \dots (x-x_{n-1}). \end{aligned} \quad (6.3)$$

З практичної точки зору для визначення різниць вищих порядків використовують вираз:

$$\Delta^j y_i = \Delta(\Delta^{j-1} y_i) = \Delta^{j-1} y_{i+1} - \Delta^{j-1} y_i.$$

При $n=1$ з (6.3) отримуємо формулу для лінійної інтерполяції

$$P_1(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0),$$

а якщо $n=2$ – формулу параболічної чи квадратичної інтерполяції

$$P_2(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2h^2}(x - x_0)(x - x_1).$$

Якщо дана необмежена кількість значень функції x , то кількість n може бути будь-якою. Практично в цьому випадку n вибирають так, щоб різниця $\Delta^n y_i$ була постійною з заданою точністю. За початкове значення x_0 можна приймати будь-яке табличне значення аргументу x . Коли кількість значень функції скінченна, то кількість n обмежена та не може бути більше кількості значень функції y , зменшеної на одиницю.

Формула (6.3) носить назву першої інтерполяційної формули Ньютона. Цей вираз незручний для інтерполяції в околі останніх значень y . В цьому випадку, як правило, використовують другу інтерполяційну формулу Ньютона, яка отримана при використанні лівих різниць від останнього значення (x_n, y_n) (інтерполяція «назад»). Тоді інтерполяційний поліном має вигляд:

$$\begin{aligned} P_n(x) = & C_0 + C_1(x - x_n) + C_2(x - x_n)(x - x_{n-1}) + \\ & + C_3(x - x_n)(x - x_{n-1})(x - x_{n-2}) + \dots \\ & \dots + C_n(x - x_n)(x - x_{n-1}) \dots (x - x_1). \end{aligned}$$

Коефіцієнти C_j визначаються таким чином:

$$\begin{aligned} C_0 &= y_n \\ C_1 &= \frac{\Delta y_{n-1}}{h} = \frac{\nabla y_n}{h}, \end{aligned}$$

∇y_n – ліва різниця першого порядку в точці y_n ,

$$C_2 = \frac{\Delta^2 y_{n-2}}{2! h^2} = \frac{\nabla^2 y_n}{2h^2},$$

$\nabla^2 y_n$ – ліва різниця другого порядку,

$$\dots\dots\dots$$

$$C_j = \frac{\Delta^j y_{n-j}}{j! h^j} = \frac{\nabla^j y_n}{j! h^j},$$

$\nabla^j y_n$ – ліва різниця j-го порядку.

Кінцевий вираз для другої інтерполяційної формули Ньютона:

$$P_n(x) = y_n + \frac{\Delta y_{n-1}}{1! h} (x - x_n) + \frac{\Delta y_{n-2}^2}{2! h^2} (x - x_n)(x - x_{n-1}) + \dots$$

$$+ \frac{\Delta^n y_0}{n! h^n} (x - x_n)(x - x_{n-1}) \dots (x - x_1).$$

Інтерполяційні формули Ньютона можуть бути використані для екстраполяції функції. Якщо $x < x_0$, то зручно використовувати першу інтерполяційну формулу Ньютона, причому

$$\frac{x - x_0}{h} < 0.$$

Якщо $x > x_n$, то використовують другу інтерполяційну формулу Ньютона, де

$$\frac{x - x_n}{h} > 0.$$

Таким чином, перша інтерполяційна формула Ньютона, як правило, використовується для інтерполяції «уперед» та екстраполяції «назад», а друга – для інтерполяції «назад» та екстраполяції «уперед».

В формулах Ньютона використовують ліві та праві різниці. Використання центральних різниць для отримання інтерполяційних формул приводить до формул Гаусса, Стірлінга та Бесселя.

Треба відзначити, що центральні різниці використовуються не в звичайному вигляді, а шляхом застосування правих різниць при поступовому зсуві індексів вліво.

Ці формули зручно розглядати на $(2n+1)$ рівновіддалених вузлах інтерполяції

$$x_{-n}, x_{-(n-1)}, \dots, x_{-1}, x_0, x_1, \dots, x_{n-1}, x_n,$$

причому

$$\Delta x_i = x_{i+1} - x_i = h \quad (i = -n, -(n-1), \dots, n-1),$$

а для функції $y = f(x)$ відомі її значення в цих вузлах $y_i = f(x_i)$.

Потрібно побудувати поліном $P(x)$ ступеня не вище $2n$ таких, що

$$P(x_i) = y_i.$$

Поліном $P(x)$ відшукується у вигляді:

$$\begin{aligned}
 P(x) = & C_0 + C_1(x-x_0) + C_2(x-x_0)(x-x_1) + C_3(x-x_{-1}) \\
 & (x-x_0)(x-x_1) + \dots + C_{2n-1}(x-x_{-(n-1)}) \dots \\
 & \dots (x-x_{-1})(x-x_0)(x-x_1) \dots (x-x_{n-1})(x-x_n).
 \end{aligned} \tag{6.4}$$

Аналогічно інтерполяційним формулам Ньютона знаходимо

$$\begin{aligned}
 C_0 &= y_0; \\
 C_1 &= \frac{\Delta y_0}{h}; \\
 C_2 &= \frac{\Delta^2 y_{-1}}{2! h^2}; \\
 &\dots \dots \dots \\
 C_{2n-1} &= \frac{\Delta^{2n-1} y_{-(n-1)}}{(2n-1)! h^{2n-1}}; \\
 &\dots \dots \dots \\
 C_{2n} &= \frac{\Delta^{2n} y_{-n}}{(2n)! h^{2n}}.
 \end{aligned}$$

Підставляючи знайдені значення коефіцієнтів в (6.4), отримаємо першу інтерполяційну формулу Гаусса, яка містить різниці (таблиця 6.1):

$$\Delta y_0, \Delta^2 y_{-1}, \Delta^3 y_{-1}, \Delta^4 y_{-2}, \Delta^5 y_{-2}, \Delta^6 y_{-2}, \dots$$

Аналогічно можна отримати другу інтерполяційну формулу Гаусса, яка містить центральні різниці:

$$\Delta y_{-1}, \Delta^2 y_{-1}, \Delta^3 y_{-2}, \Delta^4 y_{-2}, \Delta^5 y_{-3}, \Delta^6 y_{-3}, \dots$$

Використавши середнє арифметичне першої та другої інтерполяційних формул Гаусса, отримаємо формулу Стірлінга. Ці формули дозволяють вивести інтерполяційну формулу Бесселя. Взагалі, використання інтерполяційних формул з центральними різницями доцільно всередині інтервалу, тоді як по його краях, як правило, використовують формули Ньютона. Застосування цих формул розглядається в таблиці 6.1.

Похибки інтерполяції для формул Ньютона можна оцінити відповідно для першої та другої формул як:

$$\Delta_n(x) = \frac{q(q-1)\dots(q-n)}{(n+1)!} \Delta^{n+1} y_0,$$

$$\Delta_n(x) = \frac{q(q+1)\dots(q+n)}{(n+1)!} \Delta^{n+1} y_n,$$

$$\text{де } q = \frac{x - x_n}{h}.$$

Для формули Стірлінга:

$$\Delta(x_n) = \frac{\Delta^{2n+1} y_{-(n-1)} + \Delta^{2n+1} y_{-n}}{2(2n+1)!} q(q^2-1)(q^2-2^2) \dots (q^2-n^2).$$

Для випадку нерівновіддалених значень аргументу можна отримати інтерполяційні формули, вживаючи визначення поділених різниць. Наприклад, відношення

$$[x_i, x_{i+1}] = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

називається поділеною різницею першого порядку, а відношення

$$[x_i, x_{i+1}, x_{i+2}] = \frac{[x_{i+1}, x_{i+2}] - [x_i, x_{i+1}]}{x_{i+2} - x_i}$$

– поділеною різницею другого порядку.

Поділені різниці порядку n отримуються з рекурентного відношення:

$$[x_i, x_{i+2}, \dots, x_{i+n}] = \frac{[x_{i+1}, \dots, x_{i+n}] - [x_{i+1}, \dots, x_{i+n-1}]}{x_{i+n} - x_i}$$

Таблиця 6.1 – Застосування різницьових формул інтерполяції

| x | y | Δy | $\Delta^2 y$ | $\Delta^3 y$ | $\Delta^4 y$ | Примітки |
|----------|----------|-----------------|-------------------|-------------------|-------------------|-----------------------|
| | | | | | | |
| x_{-2} | y_{-2} | | $\Delta^2 y_{-3}$ | | $\Delta^4 y_{-4}$ | Друга формула Ньютона |
| | | Δy_{-2} | | $\Delta^3 y_{-3}$ | | |
| x_{-1} | y_{-1} | | $\Delta^2 y_{-2}$ | | $\Delta^4 y_{-3}$ | Формула Стрлінга |
| | | Δy_{-1} | | $\Delta^3 y_{-2}$ | | |
| x_0 | y_0 | | $\Delta^2 y_{-1}$ | | $\Delta^4 y_{-2}$ | Формула Бесселя |
| | | Δy_0 | | $\Delta^3 y_{-1}$ | | |
| x_1 | y_1 | | $\Delta^2 y_0$ | | $\Delta^4 y_{-1}$ | Перша формула Ньютона |
| | | Δy_1 | | $\Delta^3 y_0$ | | |
| x_2 | y_2 | | $\Delta^2 y_1$ | | $\Delta^4 y_0$ | |
| | | Δy_2 | | $\Delta^3 y_1$ | | |
| x_3 | y_3 | | $\Delta^2 y_2$ | | $\Delta^4 y_1$ | |
| | | | | | | |

Можна отримати інтерполяційну формулу Ньютона для нерівновіддалених значень аргументу:

$$P(x) = y_0 + [x_0, x_1](x - x_0) + [x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + [x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \dots (x - x_{n-1}).$$

6.1.2 Інтерполяція за Лагранжем

Інтерполяція за Лагранжем (Lagrange interpolation) вживається в загальному випадку для довільно розташованих вузлів.

Інтерполяційний поліном для методу Лагранжа поданий у вигляді:

$$P_n(x) = y_0 b_0(x) + y_1 b_1(x) + \dots + y_n b_n(x),$$

де всі $b_j(x)$ ($j=0, \dots, n$) – поліноми ступеня n , коефіцієнти яких можна знайти з допомогою $(n+1)$ рівняння:

$$P_n(x_i) = y_i,$$

внаслідок чого отримуємо систему рівнянь:

$$\begin{aligned} y_0 b_0(x_0) + y_1 b_1(x_0) + \dots + y_n b_n(x_0) &= y_0; \\ \dots & \\ y_0 b_0(x_n) + y_1 b_1(x_n) + \dots + y_n b_n(x_n) &= y_n. \end{aligned}$$

Якщо значення $b_j(x_i)$ обирається так, що

$$b_j(x_i) = \begin{cases} 1, & \text{при } i = j, \\ 0, & \text{при } i \neq j, \end{cases}$$

то записана система рівнянь буде задовільна.

Ця умова означає, що будь-який поліном $b_j(x)$ дорівнює нулю при кожному x_i , крім рівного x_j . Тому в загальному випадку поліном $b_j(x)$ має такий вигляд:

$$b_j(x) = C_j (x - x_0)(x - x_1) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_n).$$

Якщо $b_j(x_j) = 1$, то коефіцієнти C_j визначаються з виразу:

$$C_j = 1 / (x_j - x_0) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_n).$$

Для полінома, який шукаємо, отримуємо:

$$P_n(x) = \sum_{j=0}^n y_j * \frac{(x - x_0)(x - x_1) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_n)}{(x_j - x_0)(x_j - x_1) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_n)}.$$

Вводячи позначення

$$L_j(x) = (x - x_0)(x - x_1)\dots(x - x_{j-1})(x - x_{j+1})\dots(x - x_n),$$

записуємо

$$P_n(x) = \sum_{j=0}^n y_j \frac{L_j(x)}{L_j(x_j)}.$$

Треба відзначити дві головні властивості поліномів Лагранжа:

1) $\sum_{j=0}^n \frac{L_j(x)}{L_j(x_j)} = 1;$

2) якщо $P_n(x)$ лінійно залежить від y_j , то слушний принцип суперпозиції: інтерполяційний поліном суми декількох функцій дорівнює сумі інтерполяційних поліномів доданків.

Залишковий член інтерполяційної формули.

Припускаючи вузли інтерполяції відмінними один від другого, а функцію $f(x)$ такою, що має неперервну похідну порядку $n+1$ на проміжку $[a, b]$, де розміщені вузли інтерполяції, можна записати залишковий член інтерполяційної формули

$$R_n(x) = f(x) - P_n(x)$$

на цьому проміжку у вигляді

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1}(x),$$

де

$$\xi \in [\alpha_1, \alpha_2], \alpha_1 = \min(x, x_0, x_1, \dots, x_n), \alpha_2 = \max(x, x_0, x_1, \dots, x_n).$$

Тоді

$$|R_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |\omega_{n+1}(x)|,$$

де

$$M_{n+1} = \max_{x \in [\alpha_1, \alpha_2]} |f^{(n+1)}(x)|.$$

6.1.3 Сплайн-інтерполяція

Інтерполяція багаточленом Лагранжа або Ньютона на всьому відрізку з використанням великого числа вузлів інтерполяції часто призводить до неточного наближення, що пояснюється сильним накопиченням похибок в процесі обчислень. Крім того, через розбіжність процесу інтерполяції збільшення числа вузлів не завжди приводить до підвищення точності. Для того, щоб уникнути великих похибок, увесь відрізок розбивають на часткові відрізки і на кожному з часткових відрізків приблизно замінюють функцію $f(x)$ багаточленом невисокого ступеня (так звана кусково-поліноміальна інтерполяція).

Одним із способів інтерполяції на всьому відрізку є інтерполяція за допомогою сплайн-функцій. *Сплайн-функцією* або *сплайном* називають кусково-поліноміальну функцію, що визначена на відрізку і має на цьому відрізку деяке число неперервних похідних.

Слово «сплайн» (англійське «spline») означає гнучку лінійку, що використовується для проведення гладких кривих через задані точки площини. Основною перевагою сплайнів є можливість локально змінювати форму кривої на виділеному проміжку значень.

Коли відрізок $[a; b]$ досить великий, то не можна підвищувати точність інтерполяції за рахунок збільшення порядку інтерполяційного полінома. Це пов'язано з тим, що у полінома n -го порядку може бути $n-1$ точка екстремуму. При $n \rightarrow \infty$ графік полінома починає сильно коливатись і $|f(x) - P_n(x)|$ не наближається до нуля. Таке явище називають *феноменом Рунге*.

Тому більш перспективним є застосування кусково-поліноміальної інтерполяції, при якій апроксимувальна функція складається з окремих поліномів, як правило, однакового невеликого порядку, визначених кожен на своїй частині відрізка $[a; b]$.

Розглянемо інтерполяцію *кусово-лінійною* та *кусово-квадратичною* функціями.

Нехай $f(x)$ задана на системі вузлів $a \leq x_0 < x_1 < \dots < x_n \leq b$ й необхідно для розв'язання задачі інтерполяції побудувати кусково-лінійну функцію $\varphi(x)$, виходячи з умови критерію $\varphi(x_i) = y_i = f(x_i)$, $i = \overline{0, n}$.

Кусово-лінійною функцією (лінійним сплайном) називається функція:

$$\varphi(x) = \begin{cases} a_1x + b_1, & x \in [x_0, x_1] \\ a_2x + b_2, & x \in [x_1, x_2] \\ \dots & \dots \\ a_nx + b_n, & x \in [x_{n-1}, x_n] \end{cases}$$

Для знаходження n пар її коефіцієнтів маємо систему з $2n$ лінійних рівнянь:

$$\left\{ \begin{array}{l} a_1 x_0 + b_1 = y_0 \\ a_1 x_1 + b_1 = y_1 \\ a_2 x_1 + b_2 = y_1 \\ a_2 x_2 + b_2 = y_2 \\ \dots \\ a_n x_{n-1} + b_n = y_{n-1} \\ a_n x_n + b_n = y_n \end{array} \right.$$

Графік кусково-лінійної функції має вигляд (рис. 6.2):

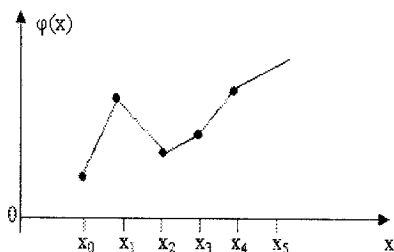


Рисунок 6.2 – Графік кусково-лінійної функції

Кусково-лінійна функція $\varphi(x)$ всередині інтервалу $(x_{i-1}; x_i)$, $i = \overline{1, n}$ неперервна й диференційовна, а в точках x_i , $i = \overline{0, n-1}$ неперервна, але не диференційовна.

Розглянемо кусково-квадратичну функцію.

Нехай $f(x)$ задана таблично на відрізку $[a; b]$, але $n=2m$ (парне) $a \leq x_0 < x_1 < \dots < x_n \leq b$.

Кусково-квадратичною функцією (квадратичним сплайном) називається функція:

$$\varphi(x) = \begin{cases} a_1 x^2 + b_1 x + c_1, & x \in [x_0, x_2] \\ a_2 x^2 + b_2 x + c_2, & x \in [x_2, x_4] \\ \dots \\ a_m x^2 + b_m x + c_m, & x \in [x_{2m-2}, x_{2m}] \end{cases}$$

Для пошуку невідомих коефіцієнтів a_k , b_k , c_k , $k = \overline{1, m}$ формується система рівнянь на основі критерію інтерполяції $\varphi(x_i) = y_i = f(x_i)$, $i = \overline{0, n}$. Ця система складається з m систем:

$$\begin{cases} a_k x_{2k-2}^2 + b_k x_{2k-2} + c_k = y_{2k-2} \\ a_k x_{2k-1}^2 + b_k x_{2k-1} + c_k = y_{2k-1} \\ a_k x_{2k}^2 + b_k x_{2k} + c_k = y_{2k} \end{cases} .$$

Наприклад, при $k=1$:

$$\begin{cases} a_1 x_0^2 + b_1 x_0 + c_1 = y_0 \\ a_1 x_1^2 + b_1 x_1 + c_1 = y_1 \\ a_1 x_2^2 + b_1 x_2 + c_1 = y_2 \end{cases} .$$

Графічно ця функція має вигляд (рис. 6.3):

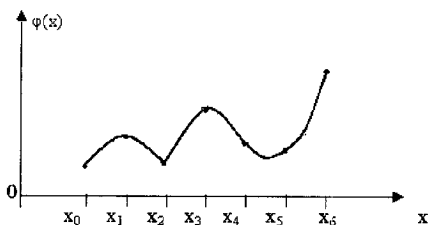


Рисунок 6.3 – Графік кусково-квадратичної функції

Кусково-квадратична функція всередині кожного інтервалу (x_{i-2}, x_i) є неперервною та диференційовною двічі, а в точках x_i неперервна, але недиференційовна.

Узагальненням розглянутих функцій є сплайни.

Нехай на відрізку $[a; b]$ задана система вузлів $a_0 \leq x_0 < x_1 < \dots < x_n \leq b$.

Сплайном $S_m(x)$ називається функція, яка визначена на відрізку $[a; b]$, належить класу l раз неперервно-диференційованих функцій й така, що на кожному відрізку $[x_{k-1}; x_k]$, $k = \overline{1, n}$ вона є поліномом порядку m .

Різницю $(m-l)$ називають *дефектом сплайна*, що показує різницю між порядком поліномів, які його складають, і ступенем гладкості загальної функції.

Якщо сплайн побудований за деякою функцією $f(x)$ таким чином, що $S_m(x_i) = f(x_i)$, то сплайн називають *інтерполяційним*. Вузли сплайна й вузли інтерполяції функції можуть не збігатися.

Очевидно, що кусково-лінійна функція є інтерполяційним сплайном порядку 1 й дефекту 1, а кусково-квадратична функція є інтерполяційним сплайном порядку 2, дефекту 2.

6.1.3.1 Класичний кубічний сплайн

Розглянемо найбільш відомий й поширений інтерполяційний сплайн порядку 3 дефекту 1. Ці сплайни порівняно недавно почали широко вживатися в обчислювальній математиці. В машинобудівельному кресленні вони застосовуються багато років, тому що це і є лекала чи гнучкі лінійки, які деформуються так, щоб з їх допомогою можна було провести криву через задані точки (x_i, y_i) . Можна показати (вживаючи теорію згину брусу при малих деформаціях), що сплайн – це група поєднаних кубічних багаточленів, в місцях поєднання яких перша та друга похідні неперервні. Такі функції називаються **кубічними сплайнами**. Для їх побудови необхідно задати коефіцієнти, які однозначно визначають поліном у проміжку між двома точками.

Наведемо *математичний опис кубічних сплайнів*.

Нехай на відріжку $[a, b]$ дійсної осі Ox задана сітка $a = x_0 < x_1 < \dots < x_n = b$, в вузлах якої визначені значення $y_i = f(x_i)$, $i = \overline{0, n}$ функції $f(x)$. Потрібно побудувати на відріжку $[a, b]$ неперервну функцію – сплайн $S(x)$, яка задовольняє такі вимоги.

1) На кожному відріжку $[x_{i-1}, x_i]$ сплайн $S(x)$ є багаточленом $S_i(x)$ третього степеня:

$$S_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i. \quad (6.5)$$

2) У вузлах x_i сплайн $S_i(x)$ приймає задані значення y_i , $i = \overline{0, n}$, тобто

$$\begin{aligned} S_i(x_{i-1}) &= y_{i-1}, \quad i = \overline{1, n} \\ S_i(x_i) &= y_i, \quad i = \overline{1, n}. \end{aligned} \quad (6.6)$$

Умови (6.6) потрібні для проходження сплайнів через вузли заданої сітки $a = x_0 < x_1 < \dots < x_n = b$, $y_i = f(x_i)$, $i = \overline{0, n}$. Попередні дві умови утворюють $2n$ рівнянь.

3) У внутрішніх вузлах x_i , $i = \overline{1, n-1}$ сплайн має неперервну першу і другу похідні, тобто:

$$\begin{aligned} S'_i(x_i) &= S'_{i+1}(x_i), \quad i = \overline{1, n-1}; \\ S''_i(x_i) &= S''_{i+1}(x_i), \quad i = \overline{1, n-1}. \end{aligned}$$

В точках спряження сплайнів, їх перші та другі похідні повинні бути рівними. Таких умов $2n-2$. Для знаходження сплайна потрібно знайти

коефіцієнти a_i, b_i, c_i, d_i багаточленів $S_i(x)$, $i = \overline{1, n-1}$, тобто $4n$ невідомих, які задовольняють $4n-2$ рівнянь.

Для отримання розв'язку системи потрібно два додаткових рівняння. Їх отримують, визначивши значення кривизни графіка сплайна на кінцях:

$$\begin{aligned} S''(x_0) &= k_1; \\ S''(x_n) &= k_2. \end{aligned}$$

Якщо $k_1 = k_2 = 0$, тоді такий сплайн називають природним. Коли є додаткові відомості про поведінку функції на кінцях інтервалу інтерполяції, то записуються інші крайові умови. Таким чином, кубічний сплайн, «склеєний» з кубічних парабол, проходить через задані точки, є гладеньким і має неперервну кривизну.

Для побудови кривої (6.5) знаходять чотири коефіцієнти. Запишемо вираз (6.5) у формі, що дозволяє зменшити кількість обчислювальних операцій, запропонованій Чарльзом Ермітом.

Введемо позначення:

$$h_i = x_{i+1} - x_i; \quad \omega = \frac{x - x_i}{h_i}; \quad \bar{\omega} = 1 - \omega, \quad (6.7)$$

де h_{i+1} – довжина підінтервалу; ω і $\bar{\omega}$ – допоміжні змінні; x – проміжна точка на відрізку $[x_i, x_{i+1}]$.

Штучною змінною $\omega \in [0, 1]$ виконується нормалізація змінної x на кожному відрізку інтерполювання між двома вузловими точками сітки.

$$\text{Якщо } x = x_i, \text{ то } \omega = \frac{x_i - x_i}{h_{i+1}} = 0; \text{ якщо } x = x_{i+1}, \text{ то } \omega = \frac{x_{i+1} - x_i}{h_{i+1}} = 1.$$

Тобто, коли x приймає всі значення в інтервалі $[x_i, x_{i+1}]$, змінна ω змінюється від 0 до 1, а змінна $\bar{\omega} = 1 - \omega$ змінюється від 1 до 0.

Поліноміальний сплайн третього степеня, що має неперервну першу та другу похідні на відрізку $[x_i, x_{i+1}]$, запишеться:

$$S_{i+1}(x) = \omega y_{i+1} + \bar{\omega} y_i + h_{i+1}^2 ((\omega^3 - \omega) y'_{i+1} + (\bar{\omega}^3 - \bar{\omega}) y'_i). \quad (6.8)$$

Номер сплайна збігається з індексом кінцевої точки відрізка $[x_i, x_{i+1}]$. Для запису виразу сплайна на сусідньому i -му відрізку достатньо в (6.8) зменшити всі індекси на одиницю:

$$S_i(x) = \omega y_i + \bar{\omega} y_{i-1} + h_i^2 ((\omega^3 - \omega) y'_i + (\bar{\omega}^3 - \bar{\omega}) y'_{i-1}). \quad (6.9)$$

Змінні ω і $\bar{\omega}$ визначаються відповідно до конкретного відрізка інтерполяції, тому вирази (6.8) та (6.9) містять фактично різні змінні ω і $\bar{\omega}$. Знайдемо значення сплайна $S_{i+1}(x)$ на кінцях відрізка $[x_i, x_{i+1}]$.

Маємо $x = x_i$ є початком для відрізка $[x_i, x_{i+1}]$, тому $\omega = 0$, $\bar{\omega} = 1$ і відповідно до (6.6) $S_i(x_i) = y_i$.

На кінці відрізка $[x_i, x_{i+1}]$ $\omega = 1$, $\bar{\omega} = 0$, отримаємо $S_{i+1}(x_{i+1}) = y_{i+1}$.

Для інтервалу $[x_{i-1}, x_i]$ точка x_i є кінцевою, тому $\omega = 1$, $\bar{\omega} = 0$ з формули (6.9) отримаємо $S_i(x_i) = y_i$.

Отже, виконується умова неперервності функції $S(x)$ у вузлах спряження кубічних багаточленів і функція $S(x)$ інтерполює задані значення незалежно від вибору похідних y'_i . Для зручності запису в подальших розділах зробимо заміну $y'_i = m_i$.

Для того, щоб визначити коефіцієнти m_i , $i = \overline{0, n}$, продиференціюємо (6.8) двічі як складну функцію від x , враховуючи, що $\omega' = \left(\frac{x - x_i}{h_{i+1}} \right)' = \frac{1}{h_{i+1}}$ та $\bar{\omega}' = (1 - \omega)' = -\frac{1}{h_{i+1}}$. Тоді

$$\begin{aligned} S'_{i+1}(x) &= \omega' y_{i+1} + \bar{\omega}' y_i + h_{i+1}^2 [(3\omega^2 \omega' - \omega') m_{i+1} + (3\bar{\omega}^2 \bar{\omega}' - \bar{\omega}') m_i] = \\ &= \frac{y_{i+1}}{h_{i+1}} - \frac{y_i}{h_{i+1}} + h_{i+1}^2 [(3\omega^2 - 1) \frac{m_{i+1}}{h_{i+1}} - (3\bar{\omega}^2 - 1) \frac{m_i}{h_{i+1}}] = \\ &= \Delta_{i+1} + h_{i+1} [(3\omega^2 - 1) m_{i+1} - (3\bar{\omega}^2 - 1) m_i], \end{aligned}$$

$$\text{де } \Delta_{i+1} = \frac{y_{i+1} - y_i}{h_{i+1}}.$$

Зменшуючи індекс на одиницю, отримаємо:

$$S'_i(x) = \Delta_i + h_i ((3\omega^2 - 1) m_i - (3\bar{\omega}^2 - 1) m_{i-1}) \quad (6.10)$$

Для сплайна (6.9) виконується умова рівності другої похідної у внутрішніх точках інтерполяційної сітки. Тому для знаходження невідомих коефіцієнтів m_i записують систему рівнянь $S''_i(x) = S''_{i+1}(x)$, $i = \overline{0, n-1}$, використовуючи (6.10). Однозначний розв'язок системи отримаємо введенням додаткових крайових умов.

6.1.3.2 Створення сплайнових параметричних кривих

Особливість сплайнових параметричних кривих полягає у вираженні невідомих коефіцієнтів сплайна через похідні у визначених точках кривої. Такий прийом зменшує кількість невідомих полінома. Також потрібно відмітити простоту переходу запису параметричних кривих від одного до багатьох вимірів.

Криві і поверхні можуть бути подані явно, неявно і параметрично. Параметричне подання дуже широко застосовується в комп'ютерній графіці через простоту та універсальність представлення. В наступних підрозділах наведено приклади знаходження невідомих коефіцієнтів та безпосередньо самих параметричних кривих.

Нехай необхідно побудувати кубічну параметричну криву, визначену трьома точками та тангенсом нахилу кривої в середній точці (рис. 6.4).

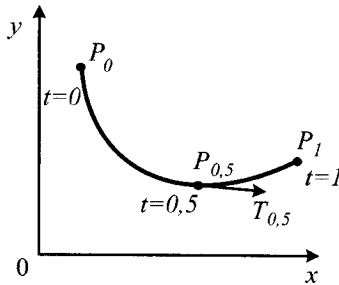


Рисунок 6.4 – Типова крива при побудові сплайна за трьома точками і значенням похідної в середній точці

Спершу побудуємо криву для змінної $x(t)$ (рис. 6.5). На криву накладається чотири обмеження для знаходження всіх чотирьох параметрів.

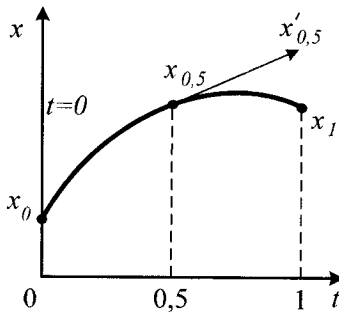


Рисунок 6.5 – Побудова кривої для змінної $x(t)$

$$\begin{cases} x(0) = [0 & 0 & 0 & 1] \cdot A; \\ x(1) = [1 & 1 & 1 & 1] \cdot A; \\ x(0,5) = [0,5^3 & 0,5^2 & 0,5 & 1] \cdot A; \\ x'(0,5) = [3 \cdot 0,5^3 & 2 \cdot 0,5^2 & 1 & 0] \cdot A. \end{cases} \quad (6.11)$$

Запишемо рівняння (6.11) в матричному вигляді.

$$G_x = \text{Coef} \cdot A;$$

$$G_x = \begin{bmatrix} x(0) \\ x(1) \\ x(0,5) \\ x'(0,5) \end{bmatrix} \text{Coef} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0,125 & 0,25 & 0,5 & 1 \\ 0,75 & 1 & 1 & 0 \end{bmatrix}. \quad (6.12)$$

З рівняння (6.12) досить просто знаходимо матрицю невідомих коефіцієнтів A :

$$A = G_x \cdot \text{Coef}^{-1}. \quad (6.13)$$

Підставивши (6.12) у вираз (6.13), отримаємо:

$$x(t) = T \cdot G_x \cdot \text{Coef}^{-1}. \quad (6.14)$$

Матрицю Coef^{-1} зазвичай називають базисною. Для зручності запису виконаємо заміну $\text{Coef}^{-1} = M$. В нашому випадку матриця M дорівнює:

$$M = \begin{bmatrix} -4 & 4 & 0 & -4 \\ 8 & -4 & -4 & 6 \\ -5 & 1 & 4 & -2 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Помноживши матрицю T та матрицю M , отримаємо сім'ю базових функцій $f1(x), \dots, f4(x)$. Вони виконують роль основних складових компонентів для створення геометричних утворень з різними ваговими коефіцієнтами. Тобто,

$$\begin{aligned} x(t) &= [f1(t) \quad f2(t) \quad f3(t) \quad f4(t)] \cdot G_x; \\ f1(t) &= -4t^3 + 8t^2 - 5t + 1; \\ f2(t) &= -4t^2 + 4t; \\ f3(t) &= -4t^3 + 6t^2 - 2t; \\ f4(t) &= 4t^3 - 4t^2 + 1. \end{aligned} \quad (6.15)$$

Зобразимо функції (6.15) на рис. 6.6.

Невідомі коефіцієнти для змінних $y(t)$ та $z(t)$ знаходяться аналогічно. Базова матриця та базові функції залишаються незмінними, змінюється лише геометричний вектор G . Тому загальне параметричне рівняння матиме вигляд:

$$x(t) = T \cdot M \cdot G_x;$$

$$y(t) = T \cdot M \cdot G_y;$$

$$z(t) = T \cdot M \cdot G_z;$$

$$P(x) = T \cdot M \cdot G.$$

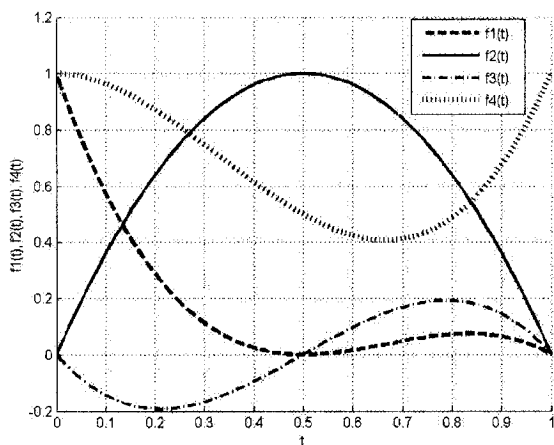


Рисунок 6.6 – Базові функції

В цьому прикладі розглядається побудова ермітової кривої (Hermitian curve) (рис. 6.7), яка задається координатами двох крайових точок та значеннями їх похідних (дотичних).

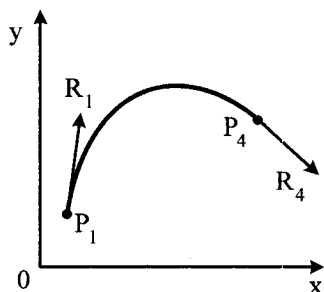


Рисунок 6.7 – Приклад побудови ермітової кривої

Визначимо рівняння для побудови ермітової кривої через геометричний вектор: $G_b = [P_1 \ P_4 \ R_1 \ R_4] \cdot T$. Виконаємо дії, аналогічні попередньому пункту. Запишемо систему рівнянь для знаходження невідомих коефіцієнтів:

$$\begin{cases} x(0) = [0 \ 0 \ 0 \ 1] \cdot A; \\ x(1) = [1 \ 1 \ 1 \ 1] \cdot A; \\ x'(0) = [0 \ 0 \ 1 \ 0] \cdot A; \\ x'(1) = [3 \ 2 \ 1 \ 0] \cdot A. \end{cases} \quad (6.16)$$

В матричному вигляді рівняння (6.16) запишеться $G_x = Coef \cdot A$. Звідки коефіцієнт A розраховується як $A = G_x \cdot M$, $Coef^{-1} = M$. В нашому випадку матриця M дорівнює

$$M = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Подібно до попереднього прикладу розрахуємо параметричну криву, помноживши матрицю T на матрицю M . Отримаємо сім'ю базових функцій $f1(x), \dots, f4(x)$:

$$\begin{aligned} x(t) &= [f1(t) \ f2(t) \ f3(t) \ f4(t)] \cdot G_x; \\ f1(t) &= 2t^3 - 3t^2 + 1; \\ f2(t) &= -2t^3 + 3t^2; \\ f3(t) &= t^3 - 2t^2 + t; \\ f4(t) &= t^3 - t^2. \end{aligned} \quad (6.17)$$

Зобразимо функції (6.17) на рис. 6.8.

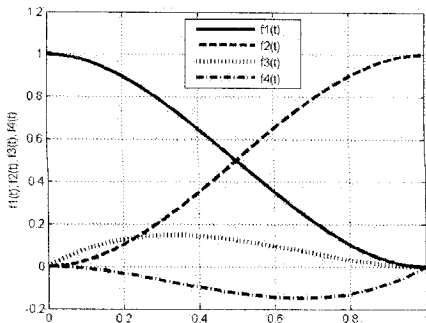


Рисунок 6.8 – Базові ермітові функції

Криві Без'є (Bezier Curves) – головний елемент побудови криволінійних форм у всіх без винятку програмах комп'ютерної графіки; за їх допомогою можна досить точно апроксимувати будь-яку лінію змінної кривизни.

Крива Без'є – це один з найпоширеніших видів гладких параметричних кривих. Ці криві знайшли широке застосування в сучасних CAD-системах. Також вони стали невід'ємною частиною операційної системи Windows (наприклад, вони використовуються для відображення шрифтів), і тому функція побудови кривих Без'є входить в стандартний набір функцій Windows GDI+. Криві Без'є за своєю суттю є варіацією ермітових кривих. Вони вказані чотирма точками (рис. 6.9).

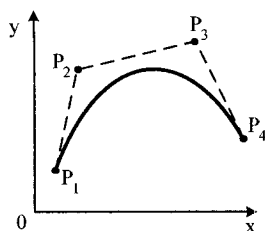


Рисунок 6.9 – Крива Без'є

Визначимо рівняння для побудови ермітової кривої через геометричний вектор

$$G_h = [Ph_1 \quad Ph_4 \quad Rh_1 \quad Rh_4] \cdot T; \quad \begin{cases} Ph_1 = P_1; \\ Ph_4 = P_2; \\ Rh_1 = 3(P_2 - P_1); \\ Rh_4 = 3(P_4 - P_3). \end{cases} \quad (6.18)$$

Виконаємо дії, аналогічні до попереднього пункту. Запишемо систему рівнянь для розрахунку невідомих коефіцієнтів:

$$\begin{bmatrix} Ph_1 \\ Ph_4 \\ Rh_1 \\ Rh_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

Загальне параметричне рівняння запишеться у вигляді:

$$P(t) = T \cdot M \cdot Pb,$$

$$\text{де } Pb = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}; \quad M = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Сім'я базових функцій $f_1(x), \dots, f_4(x)$ для кривих Без'є записується:

$$\begin{aligned} P(t) &= f_1(t) \cdot P_1 + f_2(t) \cdot P_2 + f_3(t) \cdot P_3 + f_4(t) \cdot P_4; \\ f_1(t) &= -t^3 + 3t^2 - 3t + 1; \\ f_2(t) &= 3t^3 - 6t^2 + 3t; \\ f_3(t) &= -3t^3 + 3t^2; \\ f_4(t) &= t^3. \end{aligned} \quad (6.19)$$

Зобразимо функції (6.19) на рис. 6.10.

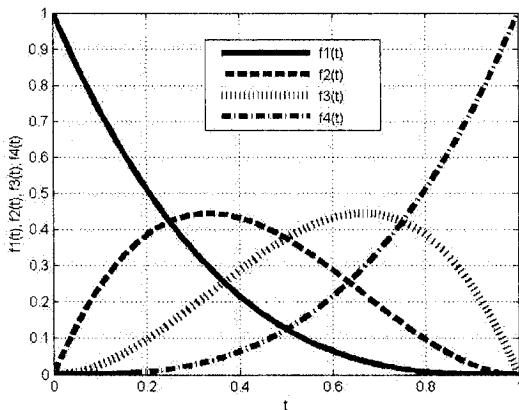


Рисунок 6.10 – Базові функції Без'є

Базові функції Без'є мають властивість опуклості. Крива ніколи не проходить через ламану, сформовану чотирма заданими вершинами. Функції Без'є задовольняють умови опуклості, а саме:

- $0 \leq f_i(t) \leq 1; t \in [0;1];$
- $f_1(t) + \dots + f_n(t) = 1.$

6.2 Апроксимація даних

Апроксимація (approximation) взагалі – це наближений опис однією функцією (апроксимувальною) заданого вигляду іншої функції (апроксимовної), яка задається у будь-якому вигляді (при апроксимації даних вона задається у вигляді масивів даних).

Існує два головних підходи до апроксимації даних. При одному з них вимагають, щоб апроксимувальна крива (можливо кусково-гладка) проходила через всі точки, які задані таблицею. Це можна зробити з допомогою методів інтерполяції, які були розглянуті в попередньому розділі. При іншому підході дані апроксимують простою функцією, яка використовується при всіх табличних значеннях, але не обов'язково, щоб вона проходила через всі точки. Такий підхід зветься припасуванням кривої, яку прагнуть провести так, щоб її відхилення від табличних даних був мінімальним. Як правило, користуються методом найменших квадратів (МНК), тобто зводять до мінімуму суму квадратів різниць між значенням функції, яка визначена обраною кривою, та таблицею.

Нехай у таблиці задана $(n+1)$ точка $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ і треба знайти апроксимувальну криву $g(x)$ в діапазоні $x_0 \leq x \leq x_n$ (рис. 6.11). В цьому випадку похибка в кожній табличній точці буде

$$\varepsilon_i = g(x_i) - y_i.$$

Тоді сума квадратів похибок визначається виразом:

$$E = \sum_{i=0}^n [g(x_i) - y_i]^2.$$

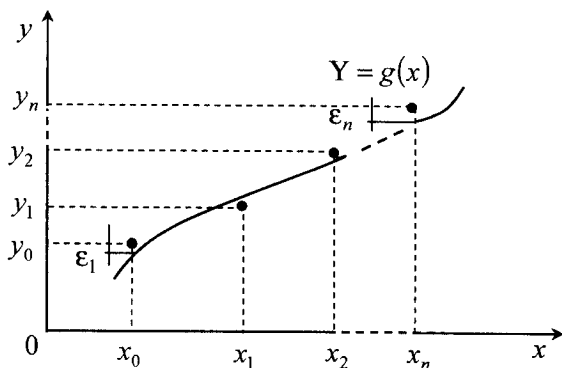


Рисунок 6.11 – Апроксимація даних

Якщо перейти до матричної форми запису, то формула методу найменших квадратів матиме вигляд:

$$C = [U^T \cdot U]^{-1} \cdot U^T \cdot Y,$$

$$\text{де } C = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_k \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, U = \begin{bmatrix} g_1(x_1) & \dots & g_k(x_1) \\ g_1(x_2) & \dots & g_k(x_2) \\ & \dots & \\ g_1(x_n) & \dots & g_k(x_n) \end{bmatrix}.$$

Вибір виду функції $g(x)$ повинен здійснюватися з урахуванням характеру табличних даних (періодичності, властивості симетрії, існування асимптотики та т. п.).

Іноді таблицю розбивають на декілька частин та добирають окрему апроксимувальну криву для кожної частини. Такий підхід задовольняє ті випадки, коли дані відповідають різним фізичним станам системи.

Залишкова середня квадратична похибка апроксимації оцінюється:

$$\Delta = \sqrt{E/(n+1)}.$$

Якщо при побудові апроксимувальної функції використовуються ортогональні поліноми, для яких

$$\sum g_j(x_i)g_k(x_i) = 0, \quad \text{якщо } j \neq k,$$

то система (6.5) спрощується, і матриця стає діагональною. Коефіцієнти визначаються зі співвідношень

$$C_j = \sum_{i=0}^n g_j(x_i)y_i / \sum_{i=0}^n g_j^2(x_i).$$

Це спрощує задачу, і тому в багатьох стандартних програмах для припасування кривих використовують ортогональні поліноми.

Метод найменших квадратів може бути застосований, якщо функція $g(x)$ матиме нелінійний характер. При цьому потрібно використовувати заміни, що дозволять лінеаризувати функцію і знайти коефіцієнти за допомогою МНК. У таблиці 6.2 наведені типові лінеаризувальні заміни.

Таблиця 6.2 – Лінеаризувальні заміни до функції $y^* = Ax^* + B$

| Апроксимувальна функція | Необхідна заміна |
|----------------------------------|--|
| $y = \frac{A}{x} + B$ | $x^* = \frac{1}{x}, y^* = y$ |
| $y = \frac{D}{x+C}$ | $x^* = xy, y^* = y$ $C = -\frac{1}{A}, D = -\frac{B}{A}$ |
| $y = \frac{1}{Ax+B}$ | $x^* = x, y^* = \frac{1}{y}$ |
| $y = \frac{x}{Ax+B}$ | $x^* = \frac{1}{x}, y^* = \frac{1}{y}$ |
| $y = A \ln x + B$ | $x^* = \ln x, y^* = y$ |
| $y = C \cdot e^{A \cdot x}$ | $x^* = x, y^* = \ln y, C = e^B$ |
| $y = C \cdot x^A$ | $x^* = \ln x, y^* = \ln y, C = e^B$ |
| $y = (Ax+B)^{-2}$ | $x^* = x, y^* = y^{-\frac{1}{2}}$ |
| $y = C \cdot x \cdot e^{-Dx}$ | $x^* = x, y^* = \ln \frac{y}{x}, C = e^B, D = -A$ |
| $y = \frac{L}{1+C \cdot e^{Ax}}$ | $x^* = x, y^* = \ln \left(\frac{y}{L} - 1 \right), C = e^B$ |

6.3 Статистична обробка даних

При обробці результатів експериментів у вимірjuвальній техніці (наприклад, для оцінювання похибки вимірювань), автоматичі (розв'язування задач ідентифікації, оптимального управління), статистичній радіотехніці виникає необхідність оцінювання характеристик випадкової величини.

Як оцінка \bar{x} невідомого математичного сподівання m_X випадкової величини X використовується середнє арифметичне результатів n незалежних випробувань

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n},$$

а для оцінки дисперсії σ_X^2 – співвідношення

$$\overline{\delta^2} = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}.$$

При припущенні нормального закону розподілу величини x можна показати, що величина

$$T = \frac{\bar{x} - m_x}{\overline{\delta} / \sqrt{n}}$$

має t -розподіл Стюдента з $k = n - 1$ ступенями свободи. Звідси можна визначити надійний інтервал для справжнього значення x : за відомими значеннями надійної імовірності P з таблиці 6.3 знаходимо ε .

Таблиця 6.3 – Значення для інтервалу $-\varepsilon < t < \varepsilon$, де величина t має розподіл Стюдента в залежності від надійної імовірності p і числа ступенів вільності k

| k | $P=0,90$ | $P=0,95$ | $P=0,99$ |
|-----|----------|----------|----------|
| 1 | 6,310 | 12,71 | 63,7 |
| 2 | 2,920 | 4,30 | 9,92 |
| 3 | 2,350 | 3,18 | 5,84 |
| 4 | 2,130 | 2,77 | 4,60 |
| 5 | 2,020 | 2,57 | 4,03 |
| 6 | 1,943 | 2,45 | 3,71 |
| 7 | 1,895 | 2,36 | 3,50 |
| 8 | 1,860 | 2,31 | 3,36 |
| 9 | 1,833 | 2,26 | 3,25 |
| 10 | 1,812 | 2,23 | 3,17 |
| 11 | 1,796 | 2,20 | 3,11 |
| 12 | 1,782 | 2,18 | 3,06 |
| 13 | 1,771 | 2,16 | 3,01 |
| 14 | 1,761 | 2,14 | 1,98 |
| 15 | 1,753 | 2,13 | 2,95 |
| 16 | 1,746 | 2,12 | 2,92 |
| 17 | 1,740 | 2,11 | 2,90 |
| 18 | 1,734 | 2,10 | 2,86 |
| 19 | 1,729 | 2,09 | 2,86 |
| 20 | 1,725 | 2,08 | 2,84 |
| 22 | 1,717 | 2,07 | 2,82 |
| 24 | 1,711 | 2,06 | 2,80 |
| 26 | 1,706 | 2,06 | 2,78 |
| 28 | 1,701 | 2,05 | 2,76 |
| 30 | 1,697 | 2,04 | 2,75 |
| 40 | 1,684 | 2,02 | 2,70 |
| 60 | 1,671 | 2,00 | 2,66 |
| 120 | 1,658 | 1,98 | 2,62 |
| 240 | 1,645 | 1,96 | 2,58 |

$$\Delta = \varepsilon * \frac{\overline{\delta^2}}{\sqrt{n}}.$$

Таким чином, випадкова величина x розподілена за нормальним законом з математичним сподіванням m_X і дисперсією σ_X^2 . Справжнє значення x знаходиться в інтервалі $(m_X - \Delta, m_X + \Delta)$ з надійною імовірністю P .

Для оцінювання виду закону розподілу найширше застосування мають критерії Колмогорова і Пірсона, які дозволяють на основі порівняння емпіричної функції розподілу $f_X^*(x)$, одержаної у вигляді гістограми в результаті обробки експериментальних даних, з гіпотетичною $f_X(x)$, яка відповідає запропонованій гіпотезі, зробити висновки про їх збігання чи незбігання при рівні значущості α , який визначається як імовірність того, що буде відхилена достовірна гіпотеза.

В критерії Колмогорова мірою є величина

$$\lambda = \left| f_X(x) - f_X^*(x) \right|_{\max} \sqrt{n},$$

яку порівнюють з критичним значенням, заданим з таблиці 6.4.

Таблиця 6.4 – Критичні значення λ_0 в залежності від рівня значущості

| α | 0,50 | 0,40 | 0,30 | 0,20 | 0,10 | 0,05 | 0,02 | 0,001 | 0,001 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| λ_0 | 0,828 | 0,895 | 0,974 | 1,073 | 1,224 | 1,358 | 1,520 | 1,627 | 1,950 |

При $\lambda < \lambda_{\alpha p}$ гіпотеза про збігання $f_X(x)$ и $f_X^*(x)$ приймається.

В критерії Пірсона обчислюється величина

$$\chi^2 = \sum_{i=0}^k \frac{[f_X(x_i) - f_X^*(x_i)]^2}{f_X(x_i)},$$

де k – число розрядів гістограми (дискретних значень $f_X(x_i)$).

З таблиці 6.5 визначають критичне значення χ^2 , зважаючи на α і число ступенів свободи

$$r = k - l - 1,$$

де l – число параметрів, що їх містить в собі закон розподілу (для нормального $l=2$, пуасонівського $l=1$ і т. д.).

При $\chi^2 < \chi_{hp}^2$ гіпотеза приймається.

Таблиця 6.5 – Критичні точки розподілу χ^2 – випадкова величина, яка розподілена за законом χ^2 з ступенями вільності k (таблиця містить значення α , що одержані із умови $P(x) \leq \alpha$)

| Число ступенів вільності k | $\alpha = 0,01$ | $\alpha = 0,025$ | $\alpha = 0,05$ | $\alpha = 0,95$ | $\alpha = 0,975$ | $\alpha = 0,99$ |
|------------------------------|-----------------|------------------|-----------------|-----------------|------------------|-----------------|
| 1 | 6,6 | 6,0 | 3,8 | 0,0039 | 0,00098 | 0,00016 |
| 2 | 9,2 | 7,4 | 6,0 | 0,103 | 0,051 | 0,020 |
| 3 | 11,3 | 9,4 | 7,8 | 0,352 | 0,216 | 0,115 |
| 4 | 13,3 | 11,1 | 9,5 | 0,711 | 0,484 | 0,297 |
| 5 | 15,1 | 12,8 | 11,1 | 1,15 | 0,831 | 0,554 |
| 6 | 16,8 | 14,4 | 12,6 | 1,64 | 1,24 | 0,872 |
| 7 | 18,5 | 16,0 | 14,1 | 2,17 | 1,69 | 1,24 |
| 8 | 20,1 | 17,5 | 15,5 | 2,73 | 2,18 | 1,65 |
| 9 | 21,7 | 19,0 | 16,9 | 3,33 | 2,70 | 2,09 |
| 10 | 23,2 | 20,5 | 18,3 | 3,94 | 3,25 | 2,56 |
| 11 | 24,7 | 21,9 | 19,7 | 4,57 | 3,82 | 3,05 |
| 12 | 26,2 | 23,3 | 21,0 | 5,23 | 4,40 | 3,57 |
| 13 | 27,7 | 24,7 | 22,4 | 5,89 | 5,01 | 4,11 |
| 14 | 29,1 | 26,1 | 23,7 | 6,57 | 5,63 | 4,66 |
| 15 | 30,6 | 27,5 | 25,0 | 7,26 | 6,26 | 5,23 |
| 16 | 32,0 | 28,8 | 26,3 | 7,96 | 6,91 | 5,81 |

Якщо порівнюють аналітично одержані закони розподілу ймовірностей, то мірою їх близькості служить значення середньої квадратичної похибки.

Для оцінювання взаємозалежності випадкових величин, між якими існує стохастичний зв'язок, використовується коефіцієнт кореляції (correlation coefficient) (n – об'єм вибірки)

$$R_{xy} = \frac{1}{n-1} \frac{\sum_{i=1}^n (x_i - m_x)(y_i - m_y)}{\sigma_x \sigma_y}$$

При визначенні взаємозалежності значень випадкових величин в різні моменти часу коефіцієнт кореляції оцінюється за формулою

$$R_X(\tau) = \frac{1}{n-m-1} \frac{\sum_{i=1}^{n-m} [x(t_i) - m_X][x(t_i + \tau) - m_X]}{D_X},$$

де $x(t_i)$ – значення випадкової величини X в момент часу t_i , а $x(t_i + \tau)$ – в момент часу, який відрізняється від t_i на інтервал τ . Таким чином, $x(t_i) = x_i$, $x(t_i + \tau) = x_j$, τ – інтервал часу між i та j значеннями x , $i - j = m$.

Інтервал кореляції визначається як відрізок часу, за який кореляційна функція зменшується на 95%.

Обчислення кореляційної функції за відомими масивами даних x та y за допомогою наведених формул не викликає труднощів, а апроксимація виду кореляційної функції типовими кореляційними функціями (таблиця 6.6) може здійснюватися за методом найменших квадратів.

Таблиця 6.6 – Типові кореляційні функції

| Вигляд | Параметри |
|--|--|
| $R_x(\tau) = \sigma_x^2(1 - \alpha \tau),$ $\tau < 1/\alpha$ | $\alpha = (\sigma_x^2 - R_x(\tau^*)) / \sigma_x^2 \tau^*,$ $R_x(\tau^*)$ – відоме значення кореляційної функції |
| $R_x(\tau) = \sigma_x^2 e^{-\alpha \tau }$ | $\alpha = \frac{1}{\tau^*} \ln \frac{\sigma_x^2}{R_x(\tau^*)}$ |
| $R_x(\tau) = \sigma_x^2 e^{-\alpha^2 \tau^2}$ | $\alpha = \frac{1}{\tau^*} \sqrt{\ln \frac{\sigma_x^2}{R_x(\tau^*)}}$ |
| $R_x(\tau) = \sigma_x^2 e^{-\alpha \tau } (1 + \alpha \tau)$ | $\alpha \approx 4,5 / \tau_k^{max}$ |
| $R_x(\tau) = \sigma_x^2 e^{-\alpha \tau } \cos \beta \tau$ | $\alpha = \frac{\ln \frac{\sigma_x^2 \cos \frac{\pi \tau_2}{2\tau_1}}{R_x(\tau_2)}}{\tau_2^*}, \beta = \pi / 2\tau_1^*$ при двох відомих значеннях кореляційної функції $R_x(\tau)$, причому $R_x(\tau_1^*) = 0$. |

6.4 Чисельне інтегрування

В багатьох задачах, що пов'язані з аналізом, ідентифікацією, оцінювання якості різних засобів і систем автоматики та управління, виникає необхідність обчислення певних інтегралів.

Якщо функція $f(x)$ неперервна на відрізку $[a, b]$ й відома її первісна $F(x)$, то визначений інтеграл від a до b може бути обчислений за формулою Ньютона-Лейбніца

$$I = \int_a^b f(x) dx = F(b) - F(a),$$

де $F'(x) = f(x)$.

Графічно інтеграл визначається площею, що обмежується графіком функції $y = f(x)$.

Але часто точно обчислити інтеграл важко через велику складність аналітичних перетворень, а інколи це взагалі неможливо (в випадках невластних інтегралів), чи коли підінтегральна функція задана набором числових даних, наприклад, отриманих з експерименту.

Задача чисельного інтегрування (numerical integration) функції полягає в обчисленні значення визначеного інтеграла на основі ряду значень підінтегральної функції. Формули чисельного інтегрування часто називають квадратурними.

Найбільш відомими методами знаходження визначених інтегралів є:

- формули прямокутників;
- методи Ньютона-Котеса, Гаусса, Чебишова, які ґрунтуються на використанні так званих квадратурних формул, отриманих заміною $f(x)$ інтерполяційними багаточленами;
- методи Монте-Карло, ґрунтуються на використанні статистичних моделей.

6.4.1 Формули прямокутників

Ідея методу полягає в розбитті відрізка інтегрування на дрібні частини $[x_{i-1}, x_i]$ у побудові прямокутників, які спираються на відрізки $[x_{i-1}, x_i]$ й мають висоту $f(\xi_i)$. Якщо розбиття відрізка рівномірне, то $x_i = a + i \cdot h$, де h – крок:

$$h = \frac{(a-b)}{n}, \quad i = \overline{0, n}.$$

Інтеграл вважається приблизно рівним сумі площ побудованих прямокутників. *Узагальнена квадратурна формула прямокутників* має вигляд:

$$I = \int_a^b f(x) dx \approx \sum_{i=1}^n f(\xi_i)(x_i - x_{i-1}),$$

де точка $\xi_i \in (x_i, x_{i-1})$.

В залежності від вибору ξ_i розрізняють формули лівих, правих й середніх прямокутників.

Нехай $\xi_i = x_{i-1}$, **формула лівих прямокутників**, має вигляд:

$$I \approx \sum_{i=1}^n f(x_{i-1})(x_i - x_{i-1}) \text{ – для нерівновіддалених вузлів,}$$

$$I \approx h \sum_{i=1}^n f(x_{i-1}) \text{ – для рівновіддалених вузлів.}$$

Порядок точності формули – перший, $O(h)$.

Геометрична інтерпретація наведена на рис. 6.12.

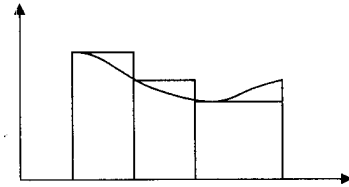


Рисунок 6.12 – Метод лівих прямокутників

Нехай $\xi_i = x_i$, **формула правих прямокутників**, має вигляд:

$$I \approx \sum_{i=1}^n f(x_i)(x_i - x_{i-1}) \text{ – для нерівновіддалених вузлів,}$$

$$I \approx h \sum_{i=1}^n f(x_i) \text{ – для рівновіддалених вузлів.}$$

Порядок точності формули – перший, $O(h)$.

Геометрична інтерпретація наведена на рис. 6.13.

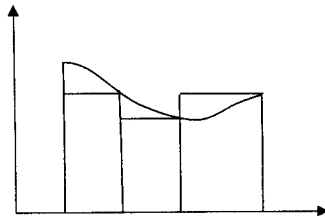


Рисунок 6.13 – Метод правих прямокутників

Нехай $\xi_i = \frac{1}{2}(x_{i-1} + x_i)$, **формула середніх прямокутників**, має вигляд:

$$I \approx \sum_{i=1}^n f\left(\frac{x_i + x_{i-1}}{2}\right)(x_i - x_{i-1}) \text{ — для нерівновіддалених вузлів,}$$

$$I \approx h \sum_{i=1}^n f\left(x_{i-1} + \frac{h}{2}\right) \text{ — для рівновіддалених вузлів.}$$

Порядок точності формул – другий, $O(h^2)$.

Геометрична інтерпретація наведена на рис. 6.14.



Рисунок 6.14 – Метод середніх прямокутників

Формули лівих та правих прямокутників можуть бути використані як для аналітично заданих функцій, так і для функцій, заданих таблично. Метод середніх прямокутників може використовуватись для пошуку інтегралів тільки від аналітично заданих функцій.

6.4.2 Формули Ньютона-Котеса

Для виведення формул Ньютона-Котеса інтеграл зображується у вигляді

$$\int_a^b f(x) dx = \sum_{i=0}^n A_i f(x_i) + \Delta, \quad (6.21)$$

де x_i – вузли інтерполяції; A_i – коефіцієнти, які залежать від вигляду формули; Δ – похибка квадратурної формули.

Замінюючи в (6.21) підінтегральну функцію відповідним інтерполяційним поліномом Лагранжа для n рівновіддалених вузлів з кроком $h = \frac{b-a}{n}$, можна одержати таку формулу для розрахунку коефіцієнтів A_i при довільній кількості вузлів.

$$A_i = \frac{b-a}{n} \frac{(-1)^{n-1}}{i!(n-1)!} \int_0^n \frac{q(q-1)\dots(q-n)}{(q-i)} dq, \quad (6.22)$$

де $q = \frac{x-a}{h}$ – приведена змінна.

Звичайно коефіцієнти $H_i = \frac{A_i}{b-a}$ називають коефіцієнтами Котеса.

При цьому формула (6.21) набуде вигляду

$$\int_a^b f(x)dx = (b-a) \sum_{i=0}^n H_i f(x_i), \quad (6.23)$$

і має такі властивості:

$$\sum_{i=0}^n H_i = 1 \quad \text{і} \quad H_i = H_{n-i}.$$

При $n=1$ і $n=2$ із (6.22) та (6.23) отримаємо формули трапецій і Сімпсона:

$$\int_a^b f(x)dx = h \left(\frac{y_0 + y_n}{2} + y_1 + y_2 + \dots + y_{n-1} \right),$$

$$\int_a^b f(x)dx = \frac{h}{3} (y_0 + y_n + 2(y_2 + y_4 + \dots + y_{n-2}) + 4(y_1 + y_3 + \dots + y_{n-1})).$$

Похибки формул трапецій і Сімпсона визначаються, відповідно, із виразів

$$\Delta = -\frac{h^3}{12} M_2 \quad \text{та} \quad \Delta = -\frac{h^5}{90} M_4,$$

де M_2 і M_4 – максимальні значення другої та четвертої похідних $f(x)$ при $x \in (a, b)$.

Складові формули Ньютона-Котеса отримаємо шляхом комбінації простих формул. Наприклад, для формул трапецій і Сімпсона (для парних n):

$$I = \frac{h}{2} [f(x_0) + 2f(x_1) + 2f(x_1) + \dots + f(x_n)],$$

$$I = \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \dots + f(x_n)].$$

Причому похибки складових формул будуть, відповідно,

$$\Delta = -n \frac{h^3}{12} M_2 \quad \text{та} \quad \Delta = -n \frac{h^5}{180} M_4.$$

Аналогічно можна отримати складові формули Ньютона-Котеса більш високих порядків.

Для оцінювання похибки на практиці можна користуватись методом Рунге (екстраполяції Річардсона), аналогічно тому, як це робиться для однокрокових методів розв'язання задачі Коші.

В таблиці 6.7 наведені значення коефіцієнтів для $n = 1, 2, \dots, 8$.

Таблиця 6.7 – Коефіцієнти Котеса

| | \hat{H}_0 | \hat{H}_1 | \hat{H}_2 | \hat{H}_3 | \hat{H}_4 | \hat{H}_5 | \hat{H}_6 | \hat{H}_7 | \hat{H}_8 | Загальний знаменник N |
|---|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|--------------------------|
| 1 | 1 | 1 | | | | | | | | 2 |
| 2 | 1 | 4 | 1 | | | | | | | 6 |
| 3 | 1 | 3 | 3 | 1 | | | | | | 8 |
| 4 | 7 | 32 | 12 | 32 | 7 | | | | | 90 |
| 5 | 19 | 75 | 50 | 50 | 75 | 19 | | | | 288 |
| 6 | 41 | 216 | 27 | 272 | 27 | 216 | 41 | | | 840 |
| 7 | 751 | 3577 | 1223 | 2989 | 2989 | 1223 | 3577 | 751 | | 17280 |
| 8 | 989 | 5888 | -928 | 10496 | -4540 | 10496 | -928 | 5888 | 989 | 28350 |

6.4.3 Формула Чебишова

Формула (6.6) може бути приведена до вигляду

$$\int_{-1}^1 f(t) dt = \sum_{i=1}^n A_i f(t_i) \quad (6.24)$$

заміною змінних

$$x = \frac{a+b}{2} + \frac{b-a}{2} t .$$

При виведенні формули Чебишова використовуються такі умови:

- коефіцієнти A_i рівні між собою;

• квадратурна формула (6.24) точна для всіх поліномів до степеня n включно.

За цих умов формула (6.24) має вигляд:

$$\int_{-1}^1 f(t) dt = \frac{2}{n} \sum_{i=1}^n f(t_i). \quad (6.25)$$

Для знаходження t_i використовуємо другу умову, згідно з якою формула (6.25) повинна бути точною для функції вигляду

$$f(t) = t^k, \quad k = 1, \dots, n.$$

Після підстановки цих функцій в (6.25) отримаємо систему рівнянь

$$\begin{cases} t_1 + t_2 + \dots + t_n = 0, \\ t_1^2 + t_2^2 + \dots + t_n^2 = \frac{n}{3}, \\ \dots \\ t_1^n + t_2^n + \dots + t_n^n = \frac{n[1 - (-1)^{n+1}]}{2(n+1)}. \end{cases} \quad (6.26)$$

Система рівнянь (6.26) має розв'язок при $n < 8$ та $n = 9$. В цій обмеженій точності і полягає недолік формули Чебишова. Значення t_i для різних n наведені в таблиці 6.8.

Таблиця 6.8 – Значення абсцис t_i в формулі Чебишова

| n | i | t_i | n | i | t_i | |
|-----|-----|-----------------|-----|----------------|----------------|----------------|
| 2 | 1;2 | $\mp 0,577350$ | 6 | 1;6 | $\mp 0,866247$ | |
| | 3 | 1;3 | | $\mp 0,707107$ | 2;5 | $\mp 0,422519$ |
| 2 | | 0 | | 3;4 | $\mp 0,266635$ | |
| 4 | 1;4 | $\mp 0,794654$ | | 7 | 1;7 | $\mp 0,883862$ |
| | 2;3 | $\mp 0,187592$ | | | 2;6 | $\mp 0,529657$ |
| 5 | 1;5 | $\mp 0,832498$ | | | 3;5 | $\mp 0,323912$ |
| | 2;4 | $\mp 0,3745413$ | 4 | | 0 | |
| | 3 | 0 | | | | |

Для довільного інтервалу (a, b) формула (6.25) приймає вигляд

$$I = \frac{b-a}{n} \sum_{i=1}^n f(x_i),$$

де
$$x_i = \frac{a+b}{2} + \frac{b-a}{2} t_i.$$

Похибка обчислень за методом Чебишова:

$$\Delta = \int_a^b \frac{\left(x - \frac{a+b}{2}\right)^{n+1}}{(n+1)!} f^{(n+1)}(x) dx - \frac{b-a}{n(n+1)!} \sum_{i=1}^n \left(x_i - \frac{a+b}{2}\right)^{n+1} f^{(n+1)}(x).$$

6.4.4 Формула Гаусса

Формула Гаусса називається формулою найвищої алгебраїчної точності. Для формули вигляду (6.24) найвища точність може бути досягнута для поліномів степеня $(2n-1)$, які визначаються $2n$ постійними t_i і $A_i (i=1, 2, \dots, n)$.

Завдання полягає у визначенні коефіцієнтів A_i і абсцис точок t_i .

Для знаходження цих постійних розглянемо виконання формули (6.24) для функцій вигляду

$$f(t) = t^k, k = 0, 1, \dots, 2n-1.$$

Враховуючи, що

$$\int_{-1}^1 t^k dt = \begin{cases} 2/(k+1), \\ 0 \end{cases},$$

отримаємо систему рівнянь

$$\begin{cases} \sum_{i=1}^n A_i = 2; \\ \sum_{i=1}^n A_i t_i = 0; \\ \sum_{i=1}^n A_i t_i^2 = 1; \\ \sum_{i=1}^n A_i t_i^{2n-2} = \frac{2}{2n-1}; \\ \sum_{i=1}^n A_i t_i^{2n-1} = 0. \end{cases} \quad (6.27)$$

Ця система нелінійна, і її звичайне розв'язання пов'язане зі значними обчислювальними труднощами. Але якщо використовувати систему для поліномів вигляду

$$f(t) = t^k P_n(t), \quad k = 0, 1, \dots, n-1,$$

де $P_n(t)$ – поліном Лежандра, тоді її можна звести до лінійної відносно коефіцієнтів A_i з заданими точками t_i . Оскільки степені поліномів у співвідношенні не перевищують $2n-1$, повинна виконуватися система (6.27), і формула (6.24) приймає вигляд

$$\int_{-1}^1 t^k P_n(t) dt = \sum_{i=1}^n A_i t_i^k P_n(t_i). \quad (6.28)$$

В результаті властивості ортогональності ліва частина виразу (6.28) дорівнює 0, тоді

$$\sum_{i=1}^n A_i t_i^k P_n(t_i) = 0,$$

що завжди забезпечується при будь-яких значеннях A_i в точках t_i , які відповідають кореням відповідних поліномів Лежандра.

Підставляючи ці значення t_i в систему (6.27) і враховуючи перші n рівнянь, можна визначити коефіцієнти A_i .

Формула (6.24), де t_i – нулі полінома Лежандра $P_n(t)$, а $A_i, i = 1, 2, \dots, n$ визначаються із системи (6.27), називається формулою Гаусса.

Значення t_i, A_i для різних n наведені в таблиці 6.9.

Таблиця 6.9 – Елементи формули Гаусса

| n | i | t_i | A_i |
|-----|-----|-------------------|--------------------------|
| 1 | 1 | 0 | 2 |
| 2 | 1,2 | $\mp 0,57735027$ | 1 |
| 3 | 1,3 | $\mp 0,77459667$ | $\frac{5}{9}=0,55555556$ |
| | 2 | 0 | $\frac{8}{9}=0,88888889$ |
| 4 | 1,4 | $\mp 0,86113631$ | 0,34785484 |
| | 2,3 | $\mp 0,33998104$ | 0,65214516 |
| 6 | 1,6 | $\mp 0,93246951$ | 0,17132450 |
| | 2,5 | $\mp 0,66120939$ | 0,36076158 |
| | 3,4 | $\mp 0,238619119$ | 0,46791394 |
| 7 | 1,7 | $\mp 0,94910791$ | 0,12948496 |
| | 2,6 | $\mp 0,74153119$ | 0,27970540 |
| | 3,5 | $\mp 0,40584515$ | 0,38183006 |
| | 4 | 0 | 0,41795918 |
| 8 | 1,8 | $\mp 0,96028986$ | 0,10122854 |
| | 2,7 | $\mp 0,79666648$ | 0,22238104 |
| | 3,6 | $\mp 0,52553142$ | 0,31370664 |
| | 4,5 | $\mp 0,18343464$ | 0,36268378 |

Для довільного інтервалу (a,b) формула для методу Гаусса приймає вигляд

$$I = \frac{b-a}{2} \sum_{i=1}^n A_i f(x_i),$$

де

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} t_i.$$

Оцінка похибки формули Гаусса з n вузлами визначається зі співвідношення

$$\Delta \leq \frac{(b-a)^{2n+1} (n!)^4 M_{2n}}{2^{2n+1} [(2n)!]^3 (2n+1)},$$

де M_{2n} – максимальне значення $2n$ похідної на ділянці $[a, b]$.

6.4.5 Оцінювання похибки при чисельному інтегуванні

В випадку, коли підінтегральна функція задана аналітично, може бути поставлена задача про знаходження інтеграла с наперед заданою точністю. Оскільки точність розглянутих вище квадратурних формул залежить від кроку h , то точність результату можна підвищити, зменшуючи крок. Наприклад, можна ділити крок навпіл. В цьому випадку для оцінювання наближення до точного значення інтеграла використовують вже згадуваний метод Рунге:

$$I - I_{h/2} \approx \frac{I_{h/2} - I_h}{2^p - 1},$$

де I – точне значення інтеграла; I_h – наближене значення, отримане з кроком h ; $I_{h/2}$ – наближене значення, отримане з кроком $h/2$; p – порядок точності квадратурної формули.

Тому для знаходження інтеграла з заданою точністю ε потрібно обрати квадратурну формулу й початковий крок $h=h_0$. Далі розрахувати інтеграл з кроком $h/2$. Зменшувати крок навпіл необхідно до тих пір, поки не виконається умова:

$$|I_{h/2} - I_h| < \varepsilon.$$

Коли ця умова виконана, інтеграл приблизно дорівнює $I_{h/2}$.

6.4.6 Алгоритми застосування чисельних методів

Послідовність застосування формул Ньютона-Котеса

1. Вибір формули і знаходження (за допомогою таблиці 6.7) коефіцієнтів A_i .
2. Складання алгоритму та програми, причому:
 - у випадку задання дискретних значень $y_i = f(x_i)$ через крок h ці значення підставляються в (6.21);
 - у випадку задання функції $y = f(x)$ значення $y_i = f(x_i)$ вираховується, причому $x_i = x_0 + ih = a + ih (a \leq x \leq b)$.
3. Оцінювання похибки.

Послідовність застосування методу Гаусса

1. Вибір порядку методу і знаходження (за допомогою таблиці 6.9) коефіцієнтів A_i і значень t_i ($-1 \leq t \leq 1$).
2. Розбиття інтервалу $a \leq x \leq b$ на l інтервалів (рисунок 6.15).
3. Знаходження значень для кожного інтервалу ($j = 1, \dots, l$)

$$I = \sum_{j=1}^l I_j.$$

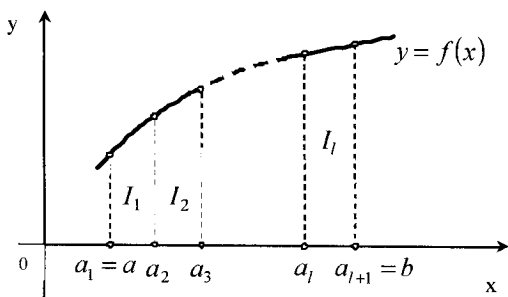


Рисунок 6.15 – Метод Гаусса. Розбиття інтервалу

При цьому значення абсцис x_i всередині кожного інтервалу j знаходяться за формулою

$$x_i = \frac{a_{j+1} + a_j}{2} + \frac{a_{j+1} - a_j}{2} t_i, \quad (6.29)$$

$$a_{j+1} = a_j + h; \quad a_1 = a;$$

де $a_{l+1} = b; \quad j = 1, \dots, l;$

$$h = \frac{b - a}{n}.$$

I_j знаходяться за формулою:

$$I_j = \int_{a_j}^{a_{j+1}} f(x) dx = \frac{(a_{j+1} - a_j)}{2} \sum_{i=1}^n A_i f(x_i). \quad (6.30)$$

4. Оцінювання похибки.

В методі Чебишова послідовність дій аналогічна методу Гаусса, але в пункті 1 коефіцієнти t_i беруться з таблиці 6.7, а в пункті 3 для знаходження j -го інтеграла використовується формула

$$I_j = \int_{a_j}^{a_{j+1}} f(x) dx = \frac{(a_{j+1} - a_j)}{n} \sum_{i=1}^n f(x_i), \quad (6.31)$$

де x_i оцінюється тим же чином, що і в методі Гаусса, за формулою (6.29).

6.4.7 Метод Монте -Карло

Метод чисельного інтегрування Монте-Карло – це найбільш відоме застосування статичного моделювання для розв'язання прикладних математичних задач.

Якщо з послідовністю випадкових чисел $\{x_i\} \in X$ з законом розподілу ймовірностей $f_x(x)$ провести функціональне перетворення

$$y_i = \varphi(x_i),$$

то математичне сподівання отриманої послідовності випадкових чисел $\{y_i\} \in Y$

$$m_Y = \int_{-\infty}^{\infty} \varphi(x) f_x(x) dx \quad (6.32)$$

при обсязі вибірки більше декількох тисяч чисел з достатньо високою точністю може бути оцінено за формулою

$$m_Y = \frac{1}{n} \sum_{i=1}^n y_i. \quad (6.33)$$

Введемо в вирази (6.29), (6.30) так звану функцію індикатора області

$$1[a, b, x] = \begin{cases} 1, & a \leq x \leq b; \\ 0, & x < a, x > b. \end{cases}$$

Якщо тепер обрати функцію

$$\varphi(x) = \frac{f(x)}{f_x(x)},$$

то кінцевий вираз буде мати вигляд

$$I = m_Y = \int_a^b f(x) dx = \frac{1}{n} \sum_{i=1}^n \frac{f(x_i)}{f_x(x_i)} [a, \sigma, x_i].$$

Алгоритм обчислення визначеного інтеграла за методом Монте-Карло наведено на рисунку 6.16.

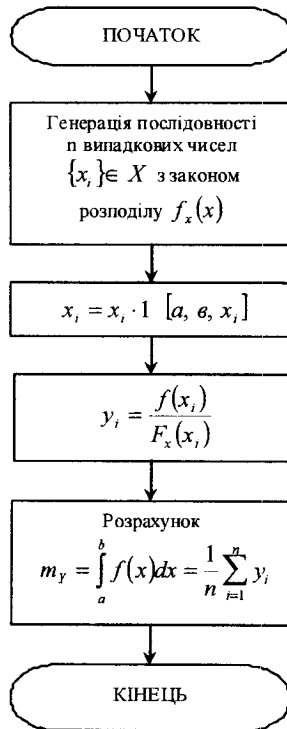


Рисунок 6.16 – Алгоритм методу Монте-Карло

Похибка методу Монте-Карло визначається похибкою генерації псевдовипадкової послідовності чисел, що згенеровані на ЕОМ, та обсягом вибірки. Вона може бути оцінена зі співвідношення

$$\Delta = \frac{1}{2\sqrt{n(1-P)}}, \quad (6.34)$$

де P – гарантована ймовірність влучання похибки в інтервал $[-\Delta; +\Delta]$.

Кількість випробувань n не залежить від кратності інтеграла, тому метод Монте-Карло знаходить застосування для обчислення багатократних

інтегралів, де застосовувати інші методи чисельного інтегрування неефективно через помітне збільшення кількості обчислювальних операцій.

Розглянемо послідовність дій при обчисленні кратних інтегралів. Для реалізації цієї процедури перш за все потрібно мати m генераторів випадкових чисел, де m дорівнює кратності інтегрованих.

Геометрично обчислення m -кратного інтеграла

$$I = \int\int\int_{(S)} \dots \int f(x_1, x_2, \dots, x_m) dx_1 dx_2 \dots dx_m, \quad (6.35)$$

де $y = f(x_1, x_2, \dots, x_m)$ – неперервна функція в обмеженій замкненій області S , зводиться до визначення $(m+1)$ -вимірному обсягу прямого циліндра в просторі $0x_1, x_2, \dots, x_m y$, що побудований на основі S й обмежений зверху поверхнею $y = f(x_1, x_2, \dots, x_m)$.

Для перетворення інтеграла (6.20) таким чином, щоб нова область інтегрування цілком знаходилась в середині одиничного m -вимірному куба σ , зробимо заміну змінних

$$x_i = a_i + (b_i - a_i)\xi_i,$$

де ξ_i – відповідні координати від 0 до 1; a_i, b_i – граничні значення координат, де розташована область інтегрування.

Тоді з (6.35) отримуємо

$$I = (a_1 - b_1)(a_2 - b_2) \dots (a_m - b_m) I_\xi,$$

де

$$I_\xi = \int\int\int_{(\sigma)} \dots \int f(\xi_1, \xi_2, \dots, \xi_m) d\xi_1 d\xi_2 \dots d\xi_m.$$

Якщо застосувати m генераторів рівномірно розподілених випадкових чисел в діапазоні $(0,1)$, то обчислення середнього значення функції від їх комбінацій з застосуванням багатовимірному індикатора області інтегрування дасть шукану оцінку інтеграла

$$I_\xi = \frac{1}{n} \sum_{i=1}^n f(\xi_{1i}, \xi_{2i}, \dots, \xi_{mi}) I[\xi_{1i}, \xi_{2i}, \dots, \xi_{mi}],$$

де $I[\xi_{1i}, \xi_{2i}, \dots, \xi_{mi}]$ дорівнює 1, якщо точка потрапляє всередину області інтегрування, і 0, якщо не потрапляє.

Похибка обчислення m -кратного інтеграла за методом Монте-Карло оцінюється аналогічно однократному за формулою (6.34).

6.5 Чисельне диференціювання

Задача полягає в тому, щоб для заданої диференційованої функції $f(x)$ знайти похідну в точці x_0 .

6.5.1 Чисельне диференціювання аналітично заданих функцій

Наближене диференціювання аналітично заданих функцій може бути потрібне при розробці універсальної процедури пошуку похідної для великої кількості різних функцій, або у випадку, коли аналітичний вигляд похідної надто громіздкий і призводить до втрати точності.

В основі чисельного диференціювання (numerical differentiation) аналітично заданих функцій покладено визначення похідної:

$$f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}.$$

Оскільки невідомо, яке значення h узяти, будується послідовність $\{h_k\}$ так, щоб $h_k \rightarrow 0$ (наприклад, $h_k = \left(\frac{1}{2}\right)^k$) і, відповідно, формується послідовність $\{D_k\}$, де

$$D_k = \frac{f(x + h_k) - f(x)}{h_k}, \quad k = 1, 2, 3, \dots, n, \dots \quad (6.36)$$

Розрахунок елементів послідовності проводять до тих пір, поки виконується умова:

$$|D_{n+1} - D_n| < |D_n - D_{n-1}|.$$

Якщо відома точність ε , з якою потрібно знайти похідну, то умова завершення може бути такою:

$$|D_{n+1} - D_n| < \varepsilon.$$

Порядок точності результату в цьому методі – h .

Нехай $f \in C^1[a, b]$. Тоді порядок точності попереднього методу можна підвищити, використовуючи замість формули (6.36) інші вирази. Розглянемо ці формули.

Формула другого порядку точності $O(h^2)$ для обчислення похідної:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

Формула отримана з наступних міркувань. Розкладемо $f(x)$ в ряд Тейлора:

$$f(x+h) = f(x) + f'(x) \cdot h + \frac{f^{(2)}(x) \cdot h^2}{2!} + \frac{f^{(3)}(C_1) \cdot h^3}{3!}.$$

$$f(x-h) = f(x) - f'(x) \cdot h + \frac{f^{(2)}(x) \cdot h^2}{2!} - \frac{f^{(3)}(C_1) \cdot h^3}{3!}.$$

Відніmemo з першої рівності другу:

$$f(x+h) - f(x-h) = 2f'(x) \cdot h + \frac{(f^{(3)}(C_1) + f^{(3)}(C_2)) \cdot h^3}{3!}$$

звідси

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$

Аналогічно можна отримати такі формули для старших похідних точності порядку $O(h^2)$:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2},$$

$$f'''(x) \approx \frac{f(x+2h) - 2f(x+h) + 2f(x-h) - f(x-2h)}{2h^3},$$

$$f^{(4)}(x) \approx \frac{f(x+2h) - 4f(x+h) + 6f(x) - 4f(x-h) + f(x-2h)}{h^4}.$$

Сам алгоритм залишається незмінним. Формується послідовність $\{h_k\}$ так, щоб $h_k \rightarrow 0$ й відповідно обчислюються елементи послідовності $\{D_{k,j}\}$, для розрахунку яких замість виразу (6.36) використовується одна з отриманих формул.

Існують формули більш високих порядків точності, їх можна знайти в спеціальній літературі.

6.5.2 Чисельне диференціювання таблично заданих функцій

При розв'язанні багатьох практичних задач виникає необхідність визначення похідних від функцій, що задаються масивами даних, отриманих з експерименту. В цьому випадку безпосереднє диференціювання може дати хибні результати, тому що не відокремлено шум, що дуже сильно впливає на похідну. Приклад зображено на рисунку 6.17, де $f(x)$ – корисний сигнал, $\tilde{f}(x)$ – сигнал, який виміряний (з шумом).

Зрозуміло, що похідна від $f(x)$ буде містити дуже сильну випадкову складову, що вносить велику похибку у визначення реальної похідної від x . Практичним шляхом розв'язання цієї проблеми є згладжування сигналу, що отриманий з експерименту, шляхом інтерполяції чи апроксимації, а далі диференціювання інтерполяційного полінома (чи апроксимувальної функції). При цьому для отримання похідних вищих порядків треба ставити умову їх збігання для сигналу та його інтерполяційного полінома. Похибка похідної інтерполяційної функції $P(x)$ дорівнює похідній від похибки цієї функції:

$$\begin{aligned}\Delta(x) &= \tilde{f}(x) - P(x), \\ \Delta^*(x) &= \tilde{f}'(x) - P'(x);\end{aligned}$$

тобто внаслідок лінійності операцій диференціювання та віднімання маємо

$$\Delta^*(x) = \Delta'(x).$$

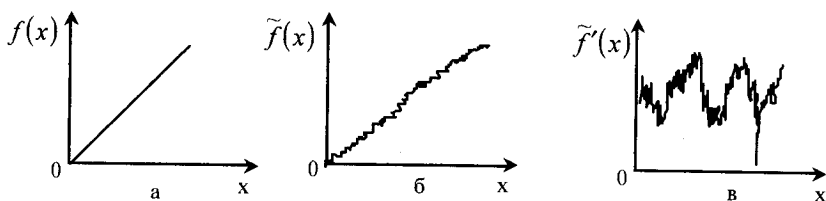


Рисунок 6.17 – Приклади корисного та виміряного сигналу

В довідниках існують спеціальні таблиці, що дозволяють знаходити з застосуванням різних різницевих інтерполяційних формул значення похідних. Взагалі такі формули нескладно отримати для будь-яких методів інтерполяції шляхом диференціювання інтерполяційних формул в загальному вигляді.

Контрольні запитання та завдання

1. Сформулюйте задачу інтерполяції. В яких випадках інтерполяція неможлива?
2. З якою точністю можна обчислити $\sqrt{2}$ за допомогою формули Лагранжа для функції \sqrt{x} , якщо вибрати вузли інтерполяції $x_0 = 81$, $x_1 = 1$, $x_2 = 4$?
3. Виведіть інтерполяційну формулу Ньютона для рівновіддалених вузлів.
4. Складіть алгоритм обчислення полінома Лагранжа.
5. Що таке сплайн-інтерполяція? Як визначаються коефіцієнти сплайнів?
6. Опишіть області застосування різницевих методів інтерполяції.
7. Що таке екстраполяція?
8. Якими шляхами розв'язується задача апроксимації?
9. В чому полягає метод Рунге?
10. Чим відрізняється апроксимація від інтерполяції?
11. Виведіть систему рівнянь для визначення коефіцієнтів апроксимувального полінома в методі найменших квадратів.
12. Наведіть формулу метода найменших квадратів.
13. Які поліноми називаються ортогональними? Наведіть приклади.
14. Що таке сплайн?
15. Розробіть алгоритм та наведіть приклад програми для ЕОМ оцінювання статистичних характеристик результатів вимірювання випадкової величини X .
16. Перелічіть методи чисельного інтегрування?
17. Чому необхідні методи і алгоритми чисельного інтегрування?
18. Яка послідовність застосування методів Ньютона-Котеса? Розробіть алгоритм. Чим відрізняються прості формули Ньютона-Котеса від складених?
19. Як отримують формули Ньютона-Котеса? Виведіть прості формули трапецій та Сімпсона для чисельного інтегрування та складені формули.
20. Як застосовуються методи Гаусса і Чебишова? Розробіть алгоритм.
21. Як визначаються коефіцієнти в формулі Гаусса для чисельного інтегрування? Чому її називають формулою найвищої алгебраїчної точності?
22. Як оцінюється похибка методів чисельного інтегрування?
23. Порівняйте методи Чебишова, Гаусса, Ньютона-Котеса?
24. Згадайте метод прямокутників. В чому його суть? Розробіть алгоритм.

25. Розробіть алгоритм та наведіть приклад програми чисельного інтегрування за методом Монте-Карло для двократного інтеграла $I = \iint_{(\sigma)} (x^2 + y^2) dx dy$, де область інтегрування σ визначається такими нерівностями

$$\frac{1}{2} \leq x \leq 1,$$

$$0 \leq y \leq 2x - 1.$$

Чи потраплять в область інтегрування точки з координатами (0.55; 0.75), (0.25; 0.75), (0.25; 0.25), (0.99; 0.70)?

Література

1. Gonzales R. C. Digital Image Processing Using MATLAB / R. C. Gonzales, R. E. Woods, S. Eddins. – Prentice Hall, Upper Saddle River, NJ, 2004. – 492 p.
2. Greenspan D. Introduction to Numerical Analysis and Applications / D. Greenspan. – Markham : Chicago, 1971. – 176 p.
3. Image Processing Toolbox For Use with Matlab, User's Guide. Version 3. – The Math Works Inc., 2004. – 775 p.
4. Квyetnyу R. Basics of Modelling and Computational Methods / R. Квyetnyу. – Вінниця : ВДТУ, 2007. – 147 с.
5. Методы компьютерной обработки изображений / Под ред. В. А. Сойфера – 2 изд. испр. – М. : ФИЗМАТЛИТ, 2003. – 784 с.
6. Абрамовиц М. Справочник по специальным функциям / Абрамовиц М., Стиган И. – М. : Наука, 1979. – 486 с.
7. Амосов А. А. Вычислительные методы для инженеров : учеб. пособ. / Амосов А. А., Дубинский Ю. А., Копченова Н. В. – М. : Высшая школа, 1994. – 554 с.
8. Анисимов Б. В. Распознавание и цифровая обработка изображений / Анисимов Б. В., Курганов В. Д., Злобин В. К. – М. : Высшая школа, 1983. – 468 с.
9. Аттетков А. В. Методы оптимизации : учеб. для вузов / Аттетков А. В., Галкин С. В., Зарубин В. С. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2003. – 440 с.
10. Батищев Д. И. Методы оптимального проектирования / Батищев Д. И. – М. : Радио и связь, 1984. – 248 с.
11. Бахвалов Н. С. Численные методы / Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. – М. : Наука, 1987. – 630 с.
12. Васильков Ю. В. Компьютерные технологии вычислений в математическом моделировании : учеб. пособие / Ю. В. Васильков, Н. Н. Василькова. – М. : Финансы и статистика, 2002. – 256 с.
13. Верлань А. Ф. Интегральные уравнения: методы, алгоритмы, программы : справочное пособие / А. Ф. Верлань, В. С. Сизиков. – К. : Наукова думка, 1986. – 544 с.
14. Вержбицкий В. М. Численные методы (математический анализ и обыкновенные дифференциальные уравнения) : учеб. пособие для вузов / Вержбицкий В. М. – М. : Высш.шк., 2001. – 382 с.
15. Воеводин В. В. Матрицы и вычисления / Воеводин В. В. – М. : Наука, 1984. – 320 с.
16. Глинченко А. С. Цифровая обработка сигналов / Глинченко А. С. – Красноярск : Изд-во КГТУ, 2001. – 199 с.

17. Глушков В. М. Основы безбумажной информатики / Глушков В. М. – М. : Наука, 1987. – 552 с.
18. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
19. Горелик А. Л. Методы распознавания / А. Л. Горелик, В. А. Скрипкин. – М. : Высшая школа, 1984. – 208 с.
20. Грузман И. С. Цифровая обработка изображений в информационных системах / И. С. Грузман, В. С. Киричук. – Новосибирск : Изд-во НГТУ, 2002. – 352 с.
21. Дахнович А. А. Дискретные системы и цифровая обработка сигналов : учебное пособие / Дахнович А. А. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2007. – 100с.
22. Демидович Б. П. Основы вычислительной математики / Б. П. Демидович, И. А. Марон. – М. : Наука, 1970. – 664 с.
23. Дубовой В. М. Основи застосування ЕОМ у інженерній діяльності / В. М. Дубовой, Р. Н. Кветний. – К. : ІСДО України, 1994. – 285 с.
24. Дубовой В. М. Програмування комп'ютеризованих систем управління та автоматики / В. М. Дубовой, Р. Н. Кветний. – Вінниця : ВДТУ, 1997. – 208 с.
25. Дубовой В. М. Програмування персональних комп'ютерів систем управління / В. М. Дубовой, Р. Н. Кветний. – Вінниця : ВДТУ, 1999.
26. Каллан Р. Основные концепции нейронных сетей / Каллан Р. – М. : Изд. дом «Вильямс», 2001. – 288 с.
27. Канторович Л. В. Оптимальные решения в экономике / Л. В. Канторович, А. Б. Горстков – М. : Наука, 1972. – 335 с.
28. Кветний Р. Н. Математическое моделирование в задачах проектирования средств автоматики и информационно-измерительной техники / Кветний Р. Н. – К. : УМК ВО, 1989. – 112 с.
29. Кветний Р. Н. Информационная теория измерений: от модели к изделию / Р. Н. Кветний, В. Т. Маликов. – М. : Знание, 1988. – 213 с.
30. Кветний Р. Н. Інтервальні моделі перетворень сигналів в інформаційно-вимірвальних системах / Р. Н. Кветний, О. Р. Бойко. – Вінниця : УНІВЕРСУМ-Вінниця, 2009. – 88с.
31. Різницеві методи та сплайни в задачах багатовимірної інтерполяції / [Кветний Р. Н., Дементьев В. Ю., Машницький М. О., Юдін О. О.]. – Вінниця : УНІВЕРСУМ-Вінниця, 2009. – 87 с.
32. Кветний Р. Н. Методи фільтрації текстурованих зображень у задачах розпізнавання та класифікації / Р. Н. Кветний, О. Ю. Софіна. – Вінниця : УНІВЕРСУМ-Вінниця, 2011. – 119 с.
33. Коллатц Л. Функциональный анализ и вычислительная математика / Коллатц Л. – М. : Мир, 1969. – 448 с.
34. Корн Г. Справочник по математике для научных работников и инженеров / Г. Корн, Т. Корн. – М. : Наука, 1986. – 832 с.

35. Краскевич В. Е. Численные методы в инженерных исследованиях / Краскевич В. Е., Зеленский К. Х., Гречко В. И. – К. : Вища школа, 1986. – 264 с.
36. Ляшенко М. Я. Чисельні методи : підручник / М. Я. Ляшенко, М. С. Головань. – К. : Либідь, 1996. – 288 с.
37. Мак-Кракен Д. Численные методы и программирование на ФОРТРАНЕ / Д. Мак-Кракен, У. Дорн. – М. : Мир, 1977. – 584 с.
38. Маликов В. Т. Вычислительные методы и применение ЭВМ / В. Т. Маликов, Р. Н. Кветный. – К. : Вища школа, 1989. – 362 с.
39. Марчук Г. И. Введение в проекционно-сеточные методы / Г. И. Марчук, В. И. Агошков. – М. : Наука, 1982. – 264 с.
40. Рабинер Л. Теория и применение цифровой обработки сигналов / Л. Рабинер, Б. Гоулд ; пер. с англ. – М. : Мир, 1979. – 578 с.
41. Ракитский Ю. В. Численные методы решения жестких систем / Ракитский Ю. В., Устинов С. М., Чернолуцкий И. Г. – М. : Наука, 1979. – 208 с.
42. Рудаков П. И. Обработка сигналов и изображений. Matlab 5.x / П. И. Рудаков, И. В. Сафонов. – М. : ДИАЛОГ-МИФИ, 2000. – 416 с.
43. Саати Т. Целочисленные методы оптимизации и связанные с ними экстремальные проблемы / Саати Т. – М. : Мир, 1973. – 302 с.
44. Самарский А. А. Введение в численные методы / Самарский А. А. – М. : Наука, 1987. – 234 с.
45. Самарский А. А. Теория разностных схем / Самарский А. А. – М. : Наука, 1977. – 400 с.
46. Скурихин В. Н. Математическое моделирование / Скурихин В. Н., Шифрин В. Б., Дубровский В. В. – К. : Техніка, 1983. – 270 с.
47. Трусов П. В. Введение в математическое моделирование : учеб. пособие / Трусов П. В. – М. : Логос, 2005. – 440 с.
48. Фисенко В. Т. Компьютерная обработка и распознавание изображений : учеб. пособие / В. Т. Фисенко, Т. Ю. Фисенко. – СПб. : СПбГУ ИТМО, 2008. – 192 с.
49. Фельдман Л. П. Чисельні методи в інформатиці : підручник / [Фельдман Л. П., Згуровський М. З., Петренко А. І., Дмитрієва О. А.]. – К. : Вид. група ВНУ, 2006. – 480 с.
50. Форсайт Дж. Машинные методы математических вычислений / Дж. Форсайт, М. Малькольм, К. Моулер. – М. : Мир, 1980. – 279 с.
51. Шуп Т. Решение инженерных задач на ЭВМ / Шуп Т. – М. : Мир, 1982. – 238 с.
52. Чабан В. Чисельні методи / Чабан В. – Львів : Вид. Нац. ун-ту "Львівська політехніка", 2001. – 186 с.
53. Яковлев А. Н. Введение в вейвлет-преобразования / Яковлев А. Н. – Новосибирск : Изд-во НГТУ, 2003. – 104 с.

Навчальне видання

**Квстний Роман Наумович
Богач Ілона Віталіївна
Бойко Олексій Романович
Софіна Ольга Юрїївна
Шушура Олексій Миколайович**

**КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ
СИСТЕМ ТА ПРОЦЕСІВ. МЕТОДИ
ОБЧИСЛЕНЬ
Частина 1**

Навчальний посібник

Редактор В. Дружиніна
Оригінал-макет підготовлено О. Софіною

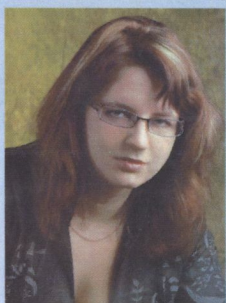
Підписано до друку 28.02.2013 р.
Формат 29,7×42¼. Папір офсетний.
Гарнітура Times New Roman.
Друк різнографічний. Ум. др. арк. 11,04.
Наклад 300 (1-й запуск – 100) прим. Зам № 2013-062.

Вінницький національний технічний університет,
комп'ютерний інформаційно-видавничий центр.
21021, м. Вінниця, Хмельницьке шосе, 95.
ВНТУ, ГНК, к. 114.
Тел. (0432) 59-85-32.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.

Віддруковано у Вінницькому національному технічному університеті
в комп'ютерному інформаційно-видавничому центрі.
21021, м. Вінниця, Хмельницьке шосе, 95.
ВНТУ, ГНК, к. 114.
Тел. (0432) 59-87-38.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.



Кветний Роман Наумович, д.т.н., професор
завідувач кафедри автоматичної та
інформаційно-вимірювальної техніки
Вінницького національного технічного
університету



Богач Ілона Віталіївна, к.т.н.,
доцент кафедри автоматичної
та інформаційно-вимірювальної
техніки, ВНТУ



Софіна Ольга Юріївна, к.т.н.,
старший викладач кафедри
автоматичної та інформаційно-
вимірювальної техніки, ВНТУ



Бойко Олексій Романович, к.т.н.,
доцент кафедри автоматичної
та інформаційно-вимірювальної
техніки, ВНТУ



Шушура Олексій Миколайович, к.т.н.,
доцент кафедри системного аналізу і
моделювання Донецького національного
технічного університету